Reduction of bivariate polynomials from convex-dense to dense, with application to factorizations

Jérémy Berthomieu, Grégoire Lecerf

To cite this version:

Jérémy Berthomieu, Grégoire Lecerf. Reduction of bivariate polynomials from convex-dense to dense, with application to factorizations. Mathematics of Computation, American Mathematical Society, 2012, 81 (279), pp.23. 10.1090/S0025-5718-2011-02562-7 . hal-00526659v2

HAL Id: hal-00526659

https://hal.archives-ouvertes.fr/hal-00526659v2

Submitted on 30 Apr 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution| 4.0 International License

REDUCTION OF BIVARIATE POLYNOMIALS FROM CONVEX-DENSE TO DENSE, WITH APPLICATION TO FACTORIZATIONS

JÉRÉMY BERTHOMIEU AND GRÉGOIRE LECERF

ABSTRACT. In this article we present a new algorithm for reducing the usual sparse bivariate factorization problems to the dense case. This reduction simply consists in computing an invertible monomial transformation that produces a polynomial with a dense size of the same order of magnitude as the size of the integral convex hull of the support of the input polynomial. This approach turns out to be very efficient in practice, as demonstrated with our implementation.

1. INTRODUCTION

Let $\mathbb{K}$ be a field. Throughout this paper, F represents the bivariate polynomial in the variables x and y over $\mathbb{K}$ that we want to factor. At the present time, the best known complexity bounds for the squarefree and irreducible factorization problems are essentially obtained in terms of the *dense size* of F. This is relevant to many situations but, in many others, it is important to take the sparsity of F into account. In this article, we present a simple method to transform F in a way that is compatible to factorizations, but so that the dense size becomes of the same order of magnitude as the size of the integral convex hull of the support of F. In the next paragraphs, we give precise definitions for the sparse and dense sizes, state our main complexity result on support reduction, and then corollaries on factorizations.

1.1. Sizes of polynomials.

Let $\mathcal{S}$ be a finite subset of points in $\mathbb{Z}^2$. The *bounding rectangle* of $\mathcal{S}$ is the smallest rectangle of the form $(o_x, o_y) + [0, d_x] \times [0, d_y]$ that contains $\mathcal{S}$, where $o_x, o_y \in \mathbb{Z}$ and $d_x, d_y \in \mathbb{N}$. We define the *dense size* of $\mathcal{S}$ as $(d_x + 1)(d_y + 1)$. We write $\operatorname{Int}\mathcal{S}$ for the *integral convex hull* of $\mathcal{S}$, that is the set of integer points inside the convex hull of $\mathcal{S}$ seen as a subset of $\mathbb{R}^2$, so that

$$\operatorname{Int}\mathcal{S} = \mathbb{Z}^2 \bigcap \left\{ \sum_{e\in\mathcal{S}} t_e e | t_e \in \mathbb{R}_{\geq 0} \text{ and} \sum_{e\in\mathcal{S}} t_e = 1 \right\}.$$

The *convex size* of $\mathcal{S}$ is defined as the cardinality $|\operatorname{Int}\mathcal{S}|$ of $\operatorname{Int}\mathcal{S}$.

For our purposes it will be convenient to consider bivariate *Laurent polynomials*. Any such polynomial $F \in \mathbb{K}[x, y, x^{-1}, y^{-1}]$ can be stored as a vector of nonzero terms, with each term composed of a coefficient and an exponent, regarded as a

Date: April 14, 2011.

2010 *Mathematics Subject Classification.* Primary 12Y05, Secondary 68W30, 11Y16, 12D05, 13P05.

Key words and phrases. Polynomial factorization.

This work has been partly supported by the French ANR-09-JCJC-0098-01 MAGIX project, and by the DIGITEO 2009-36HD grant of the Région Île-de-France.

vector in $\mathbb{Z}^2$. This storage is usually called the *sparse representation* of F. For any $(i,j) \in \mathbb{Z}^2$, we let $F_{i,j}$ denote the coefficient of $x^i y^j$ in F. The *support* of F is defined as

$$\operatorname{Supp} F = \left\{(i,j) \in \mathbb{Z}^2 | F_{i,j} \neq 0\right\}.$$

The *sparse size* of F, written σ, refers to the cardinality of the support of F. We also define the *dense size* (resp. the *convex size*) of F as the dense size (resp. convex size) of its support.

The *Newton polygon* of F, written $\operatorname{Newton} F$, is the convex hull of the support of F in $\mathbb{R}^2$. If F factors into GH, then it is known from Ostrowski [Ost21] (translated in [Ost99], and revisited later in [Ost75]) that:

$$\operatorname{Newton} F = \operatorname{Newton} G + \operatorname{Newton} H = \{a + b | a \in \operatorname{Newton} G, b \in \operatorname{Newton} H\}.$$

The latter sum of the convex hulls of G and H is usually called the *Minkowski sum*. In general, even if the sparse size of F is small compared to its convex size, the irreducible factors of F can be dense with respect to their Newton polygons, what we call *convex-dense* for short. For example, simply consider $F = y^p - x^p \in \mathbb{Q}[x,y]$, where p is a prime integer: here $\sigma = 2$ and F factors into $x - y$ and $F/(x-y)$ whose sparse size is exactly p. This shows that the irreducible factorization of F cannot be achieved in time polynomial in σ, and that the convex size of F is a relevant quantity to analyze the complexity of factorization problems.

Example 1.1. Let $F = x^{-1}y^{-1} + 1 + 2x^3 + 3y^2$. The sparse size of F is $\sigma = 4$. The Newton polygon of F is drawn in following Figure 1: the black disks represent the monomials of F, while the white disks are the other monomials contained in the Newton polygon. The convex size of F is therefore $\pi = 8$, and since the bounding rectangle of the support of F is $(-1,-1) + [0,4] \times [0,3]$, the dense size of F is 20.

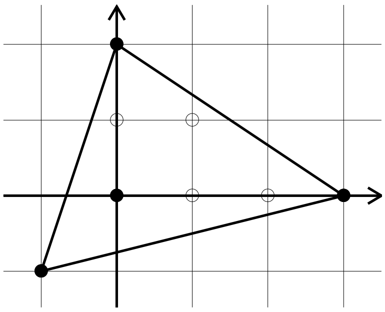

FIGURE 1. Newton polygon of $F = x^{-1}y^{-1} + 1 + 2x^3 + 3y^2$.

1.2. **Main result.** The method we propose in this paper concerns all the usual types of factorization, including the squarefree, the irreducible and the absolute ones. Our main result is a pretreatment, applied to the input polynomial, which consists of a monomial transformation that preserves the sparse size and roughly the convex size, but decreases the dense size. The monomial transformations considered are the maps of the affine group over $\mathbb{Z}^2$, written $\operatorname{Aff}\left(\mathbb{Z}^2\right)$. To be precise, these are the maps U

$$U : (i,j) \mapsto \begin{pmatrix} \alpha & \beta \\ \alpha' & \beta' \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix} + \begin{pmatrix} \gamma \\ \gamma' \end{pmatrix}, \tag{1.1}$$

with α, β, γ, α', β', and γ' in $\mathbb{Z}$, such that $\alpha\beta' - \alpha'\beta = \pm 1$. Such a map U preserves the absolute value of the volumes in $\mathbb{R}^2$.

Let $\mathcal{S}$ be a finite subset of $\mathbb{Z}^2$. The set $\mathcal{S}$ is said to be *normalized* if it belongs to $\mathbb{N}^2$ and if it contains at least one point in $\{0\} \times \mathbb{N}$, and also at least one point in $\mathbb{N} \times \{0\}$. For such a normalized set, we write d_x for the largest abscissa involved in $\mathcal{S}$ and, analogously, d_y for the largest ordinate, so that the bounding rectangle is $\mathcal{R} = [0, d_x] \times [0, d_y]$. The following theorem will be proven in Section 4.2:

Theorem 1.2. *For any normalized finite subset $\mathcal{S}$ of $\mathbb{Z}^2$, of cardinality σ, convex size π, bounding rectangle $[0, d_x] \times [0, d_y]$, and dense size $\delta = (d_x + 1)(d_y + 1)$, one can compute an invertible affine map $U \in \operatorname{Aff}(\mathbb{Z}^2)$ as in* (1.1)*, together with $U(\mathcal{S})$, with $O\left(\sigma \log^2 \delta\right)$ bit-operations, such that $U(\mathcal{S})$ is normalized of dense size at most 9π.*

Here, by the number of *bit-operations* we mean the size of the Boolean circuit that performs the computation, as in the *computation tree model* considered in [BCS97, Chapter 4]. The rest of this introduction is devoted to applications of Theorem 2 to factorizations of bivariate polynomials. Roughly speaking, given a polynomial F with a small convex size compared to its dense size, we can use the algorithm underlying Theorem 1.2 on the support of F in order to construct another polynomial to factor, with the same convex size as F but with a dense size of the same order of magnitude as the convex size. Therefore, any fast factorization algorithm in terms of the dense size leads to a fast algorithm in terms of the convex size. Another important application of our Theorem 1.2, developed by Chèze in [Chè10], concerns the decomposition of multivariate rational functions.

The proof of Theorem 1.2 is organized as follows. In our first section we explain a naive approach to reduce $\mathcal{S}$ so that the ratio of the volumes of its convex hull and of its bounding rectangle increases. The second section provides us with a uniform bound on the latter ratio reached at the end of the reduction process. The last section is then devoted to a faster dichotomic reduction algorithm, to practical performances, and to a proof that our reduction technique leads to an essentially optimal volume ratio in the worst case.

1.3. **Applications.** We shall now explain how Theorem 1.2 can be used to reduce convex-dense factorization problems to the usual dense case. For the cost analysis we use the *computation tree model* for counting the number of operations in the ground field $\mathbb{K}$. Let us recall that the "soft-Oh" notation $f(n) \in \tilde{O}(g(n))$ means that $f(n) \in g(n) \log^{O(1)}(3 + g(n))$ (we refer the reader to [GG03, Chapter 25, Section 7] for details).

If U is an affine map of $\mathbb{Z}^2$ as in (1.1), then we consider its action on the monomials, and we write $U(x^i y^j)$ for $x^{\alpha i + \beta j + \gamma} y^{\alpha' i + \beta' j + \gamma'}$. By linearity, this action extends to $\mathbb{K}[x, y, x^{-1}, y^{-1}]$ as follows:

$$U(F) = \sum_{(i,j) \in \operatorname{Supp} F} F_{i,j} U(x^i y^j).$$

Greatest common divisor. A Laurent polynomial is said to be *normalized* if its support is normalized. Let F and G be two normalized polynomials in $\mathbb{K}[x, y]$ of degree at most d_x in x and d_y in y, and with supports included in a common convex polygon of convex size π. This situation naturally occurs for instance when computing the discriminant of F, say in y, where G is set to $\frac{\partial}{\partial y} F$.

Thanks to Theorem 1.2, we can compute a reduction map U with $O\left(\sigma \log^2 \delta\right)$ bit-operations such that the partial degrees of $\tilde{F} = U(F)$ and $\tilde{G} = U(G)$ are at most $\tilde{d}_x$ in x and $\tilde{d}_y$ in y, and with $\tilde{d}_x \tilde{d}_y \in O(\pi)$. Without loss of generality we can further assume that $\tilde{d}_x \geq \tilde{d}_y$, so that the computation of $\tilde{H} = \gcd(\tilde{F}, \tilde{G})$ in $\mathbb{K}[x, y]$ can be done with $\tilde{O}\left(\pi^{1.5}\right)$ operations in $\mathbb{K}$, assuming that $\mathbb{K}$ has cardinality at least $(6\tilde{d}_y + 3)\tilde{d}_x$, by [GG03, Corollary 11.9, part *i*]. Under the same assumptions on the cardinality of $\mathbb{K}$, a *randomized* variant can also obtain the same g.c.d. with an *expected number of operations* only in $\tilde{O}(\pi)$, by [GG03, Corollary 11.9, part *ii*].

There exists a unit h in $\mathbb{K}[x, y, x^{-1}, y^{-1}]$ (that is a term $cx^i y^j$ with c invertible in $\mathbb{K}$) such that $H = hU^{-1}(\tilde{H})$ is normalized. We say that H is a *normalization* of $U^{-1}(\tilde{H})$. By the aforementioned Ostrowski theorem, it is classical to deduce that H is the actual g.c.d. of F and G, and that the convex size of H is at most π. Finally, the computation of H from $\tilde{H}$ takes $\tilde{O}(\pi \log \delta)$ more bit-operations. Of course this approach leads to a significant speedup when compared to a direct application of [GG03, Corollary 11.9] as soon as π is much smaller than δ.

Squarefree factorization. Let U be an invertible affine map over $\mathbb{Z}^2$ as in Equation (1.1), and let L be the linear part of U. Let F still be a normalized polynomial in $\mathbb{K}[x, y]$ of degree at most d_x in x and d_y in y, of sparse size σ, and of convex size π. If the squarefree factorization of F writes into $F = F_1^1 F_2^2 \cdots F_r^r$, where the F_i are the pairwise coprime squarefree factors, then

$$L(F) = L(F_1)^1 L(F_2)^2 \cdots L(F_r)^r.$$

As for the g.c.d., thanks to Theorem 1.2, we can compute a reduction map U with $O\left(\sigma \log^2 \delta\right)$ bit-operations such that the partial degrees of $\tilde{F} = U(F)$ are at most $\tilde{d}_x$ in x and $\tilde{d}_y$ in y, and with $\tilde{d}_x \tilde{d}_y \in O(\pi)$. Without loss of generality we can again assume that $\tilde{d}_x \geq \tilde{d}_y$.

If $\mathbb{K}$ has characteristic 0, then the squarefree factorization of $\tilde{F}$ takes $\tilde{O}\left(\pi^{1.5}\right)$ operations in $\mathbb{K}$ by [Lec08, Proposition 8]. This cost further drops to an expected one in $\tilde{O}(\pi)$ with the randomized variant of [Lec08, Proposition 9]. Then the squarefree factors can be easily deduced by applying U^{-1} and normalizing. Other algorithms of [Lec08] concerning the separable factorization can be also adapted in the same way to benefit of sparsity.

Irreducible factorization. If F is a Laurent polynomial, then $U(F)$ is irreducible if, and only if, F is irreducible. If F is normalized, then F is irreducible in $\mathbb{K}[x, y]$ if, and only if, F is irreducible in $\mathbb{K}[x, y, x^{-1}, y^{-1}]$. The irreducible factorization in $\mathbb{K}[x, y]$ can thus be deduced from the one in $\mathbb{K}[x, y, x^{-1}, y^{-1}]$. As for the squarefree factorization, we first compute a reduction map U, then we compute the irreducible factorization of $U(F)$, and finally we apply U^{-1} and normalize all the factors.

With this strategy, informally speaking, the algorithms of [Lec10] for instance show that the number of operations in a prime finite field can grow with only $\tilde{O}(\pi^{1.5})$. In Section 4.3 we report on examples that illustrate the speedup gained thanks to the reduction process.

1.4. **Related work.** Fast arithmetic operations on sparse polynomials are still a matter of active research. At the present time, the best performances are achieved essentially with supports being close to rectangles, thanks to the Kronecker substitution that reduces the product to a single variable [GG03, Chapter 8, Section 4].

Recent progress has been accomplished for instance in [HL10], but even when softly linear time algorithms are available for the sparse product, the overhead compared to dense sizes remains important. These facts motivate the strategy of the present paper: by a direct reduction to the dense case we avoid relying on sparse arithmetic at all.

Concerning the irreducible factorization of bivariate polynomials, the Hensel lifting and recombination technique is the most popular. It leads to the best known complexity bounds [BLS+04, Lec06, Lec07, Lec10] in the dense case. Hensel lifting is used in Bernardin's implementation within Maple [Ber97, Ber98], and in Steel's one in Magma [Ste05, BHKS09]. In order to benefit from fast Hensel lifting, which means here with a softly linear cost, in the bivariate case, one needs first to assume that F is separable, say in y, and then find a value x_0 such that $F(x_0, y)$ remains separable. Unfortunately the shift of x spoils the sparse and convex sizes. One possible solution consists in the direct computation of the irreducible factorization in $\mathbb{K}[[x]][y]$ but, at the present time, no algorithm with softly linear time is known for that task. Efforts have been made in this direction. For instance, in [AGL04] an algorithm for computing a factor of a given convex support is designed for special cases, with time polynomial in the convex size of the input polynomial. In [BHKS09], Puiseux series solutions of F are computed directly, with no shift in x. The best known complexity bounds for the Puiseux expansions seem to be found in [Pot08, PR10]. Recently, in [Wei10], Weimann proposed partial generalizations of the algorithms of [Lec06, Lec07]: if $\mathbb{K}$ is a number field, and if the polynomials supported by the exterior facets of the Newton polygon are separable, then, from their irreducible factors, one can deduce the factorization with $O(\pi^\omega)$ operations in $\mathbb{K}$, where ω is the linear algebra exponent (known to be between 2 and 2.37, but unfortunately close to 3 in practice). Compared to these methods, our approach has the advantage that it can be performed from the outset with no separability assumption, that it does not need to compute the Newton polygon, and that it can benefit from fast Hensel lifting.

Another important class of factorization algorithms is due to Gao in [Gao03], who showed that the absolute factorization can be performed in softly quadratic time in terms of the dense size. The first half of his algorithm consists in computing a basis of the first De Rham cohomology group of the complementary of the hypersurface defined by F. In [GR03] it has been shown that this task can be done in time polynomial in the convex size. When a fast sparse polynomial product is available, one can even compute the probable number of absolute factors in time softly quadratic in the convex size, over finite fields with sufficiently large characteristic [HL10, Section 7]. However these approaches still suffer an overhead when compared to the dense case, and it requires the input polynomial to be separable.

The factorization of sparse polynomials in terms of the sparse size is an active research area. Although this is not the main goal of the present article, let us mention briefly important results for multivariate polynomials. Polynomial time in terms of the sparse size of the output has been investigated by Zippel in [Zip79, Zip81] (see also [Zip93, Chapter 17]). Specifically, he proposed a probabilistic variant of Hensel lifting that runs in time polynomial in the total sparse size of the lifted factors of F in $\mathbb{K}[[x]][y]$. His results have been extended and refined in [Gat83, Kal85, GK85, Kal89]. These techniques perform well only if the lifted factors are very sparse. Finally another class of results focuses on the only computation of

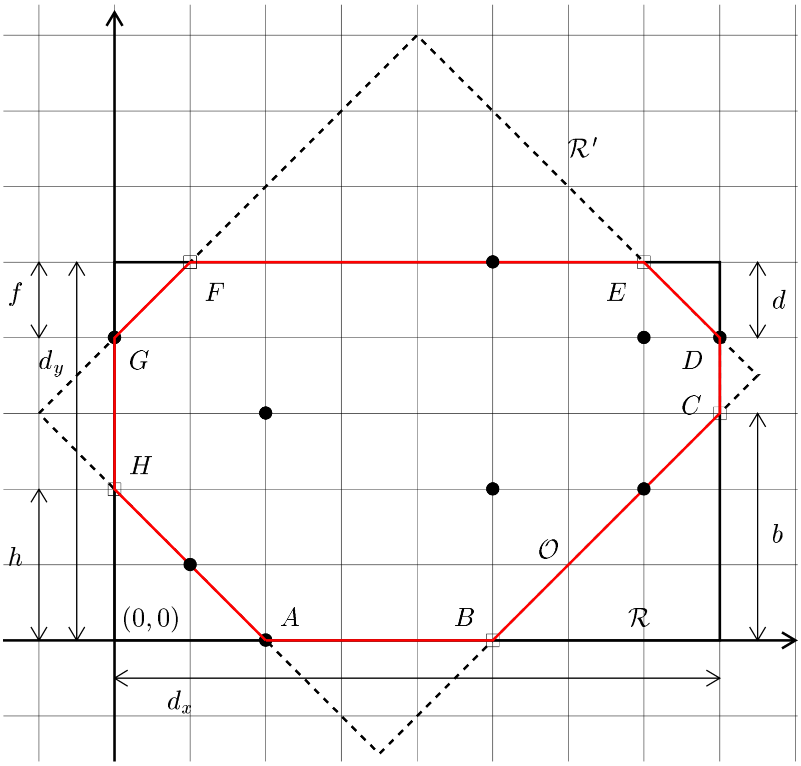

Figure 2. Bounding octagon $\mathcal{O}$ and rectangles $\mathcal{R}$ and $\mathcal{R}'$ for $\mathcal{S} = \{(2,0), (1,1), (5,2), (7,2), (2,3), (0,4), (7,4), (8,4), (5,5)\}$.

the irreducible factors of a bounded given degree. A polynomial time bound has been proved recently for this task in [AKS07] for two variables and, independently, in [KK06] directly with several variables.

2. Support reduction

This section is devoted to the reduction algorithm underlying Theorem 1.2. We start with a naive version that is to be refined in Section 4.

2.1. **Bounding rectangles.**

Let $\mathcal{S}$ be a normalized finite subset of $\mathbb{Z}^2$ with bounding rectangle $\mathcal{R} = [0, d_x] \times [0, d_y]$. We introduce the integers b, d, f and h as follows:

- $b = d_x - \max_{(i,j)\in\mathcal{S}} (i - j)$,
- $d = d_x + d_y - \max_{(i,j)\in\mathcal{S}} (i + j)$,
- $f = d_y + \min_{(i,j)\in\mathcal{S}} (i - j)$,
- $h = \min_{(i,j)\in\mathcal{S}} (i + j)$.

Then, let us define the following eight points, drawn in Figure 2 above:

$$\begin{array}{llll} A = (h, 0), & B = (d_x - b, 0), & C = (d_x, b), & D = (d_x, d_y - d), \\ E = (d_x - d, d_y), & F = (f, d_y), & G = (0, d_y - f), & H = (0, h). \end{array}$$

The rectangle $\mathcal{R}'$ supported by lines (AH), (BC), (DE), (FG) is the smallest rectangle containing $\mathcal{S}$ whose edges are parallel to the two main bissectors. The octagon $\mathcal{O} = ABCDEFGH$ contains $\mathcal{S}$ and any of its edges contains a point of $\mathcal{S}$, $\mathcal{O}$ is the *bounding octagon* of $\mathcal{S}$.

2.2. **Elementary transformations.** Our reduction algorithm will only use the three following elementary transformations. The first one, written λ, corresponds to substituting y/x into y, this yields the following map of $\mathbb{Z}^2$:

$$\begin{aligned} \lambda : \quad & \mathbb{Z}^2 \to \mathbb{Z}^2 \\ & (i,j) \mapsto (i-j, j)\,. \end{aligned}$$

We will need to swap x and y. This is the role of μ:

$$\begin{aligned} \mu : \quad & \mathbb{Z}^2 \to \mathbb{Z}^2 \\ & (i,j) \mapsto (j, i)\,. \end{aligned}$$

Finally, translations in x are necessary to normalize the supports occurring in the reduction algorithm:

$$\begin{aligned} \tau_k : \quad & \mathbb{Z}^2 \to \mathbb{Z}^2 \\ & (i,j) \mapsto (i+k, j)\,. \end{aligned}$$

2.3. **Reduced sets of points.** Applying λ to $\mathcal{S}$ modifies the volume of the bounding rectangle. For instance Figure 3 is the image of Figure 2 by λ: the height of $\mathcal{R}$ does not change, but the horizontal length becomes $d_x + d_y - b - f$. The points (i,j) in $\mathcal{S}$ that are sent to the far left of $\lambda(\mathcal{S})$ are such that $i-j$ is minimal. Analogously, those that are sent to the far right of $\lambda(\mathcal{S})$ are such that $i-j$ is maximal. Applying λ^{-1} instead of λ will result in the horizontal length of the new $\mathcal{R}$ being the difference between $\max{(i+j)}$ and $\min{(i+j)}$, namely $d_x + d_y - d - h$, as illustrated by Figure 4.

From now on and until the end of this article, η represents a real number in $[0, 3/4)$.

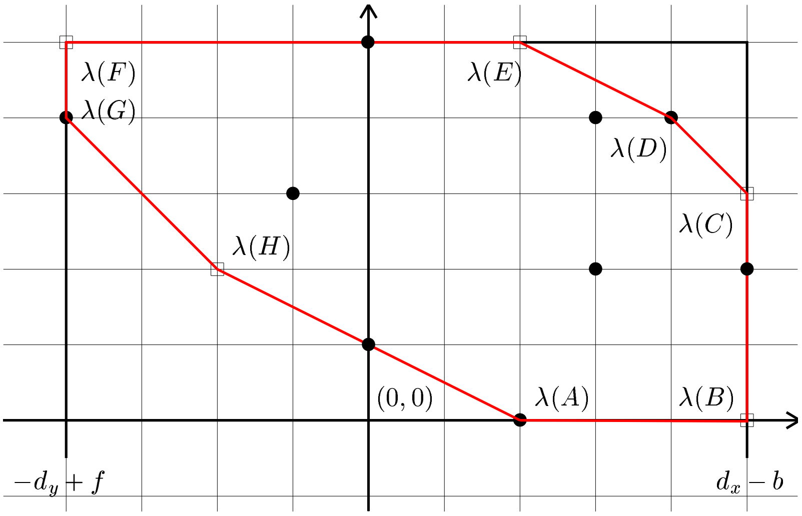

Figure 3. Image of the octagon of Figure 2 by λ, and its new bounding rectangle.

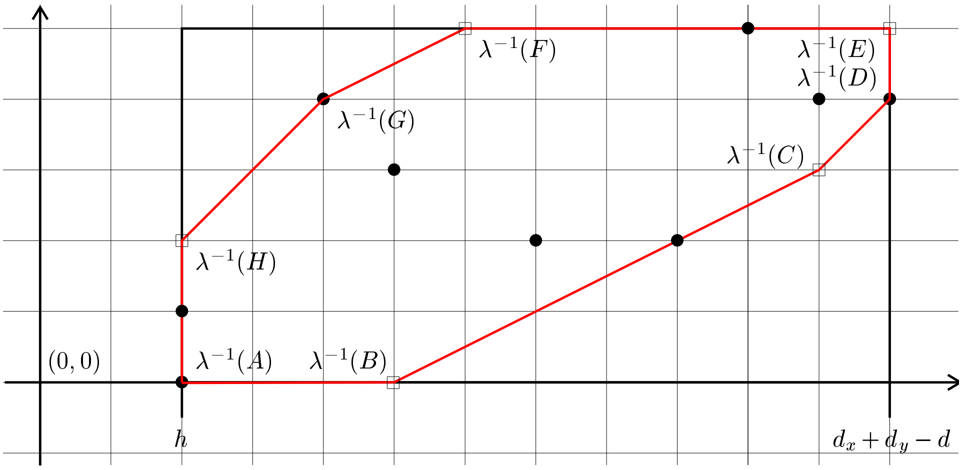

FIGURE 4. Image of the octagon of Figure 2 by λ^{-1}, and its new bounding rectangle.

Definition 2.1. A finite subset $\mathcal{S}$ of $\mathbb{Z}^2$ is said to be *η-reduced* whenever $\mathcal{S}$ is normalized, with d_x greater than or equal to d_y, and such that b, d, f and h, as defined in Section 2.1, satisfy the following two conditions:

$$b + f \leq (1+\eta) d_y \quad \text{and} \quad d + h \leq (1+\eta) d_y.$$

If $\mathcal{S}$ has only one point, then it is already η-reduced. In the next subsection, we propose an algorithm for reducing any finite subset of points of $\mathbb{Z}^2$. We shall see that η is used for controlling the tradeoff between the quality of the reduction and the time needed to reduce. The strongest reduction corresponds to $\eta = 0$.

2.4. **Degenerate case.** In this subsection we consider the case when $\mathcal{S}$ is *degenerate*, which means that all the points of $\mathcal{S}$ are aligned. If $\mathcal{S}$ is normalized and is a singleton, then it is the origin and it is already η-reduced, whatever the value of η is. Otherwise we have the following proposition:

Proposition 2.2. *For any degenerate normalized finite set of points $\mathcal{S}$ of cardinality σ, convex size π, and bounding rectangle $[0, d_x] \times [0, d_y]$, one can compute an invertible affine map $U \in \mathrm{Aff}(\mathbb{Z}^2)$ as in (1.1), together with $U(\mathcal{S})$, with $O\left(\sigma \log^2 \delta\right)$ bit-operations, where $\delta = (d_x+1)(d_y+1)$, such that:*

- *$|\alpha|$, $|\beta|$, $|\alpha'|$, and $|\beta'|$ are at most $\max(d_x, d_y, 1)$,*
- *$|\gamma|$ and $|\gamma'|$ are at most $d_x d_y$,*
- *$U(\mathcal{S})$ is normalized of dense size π.*

Proof. According to the hypotheses, the following two situations can occur: the points of $\mathcal{S}$ are either on the segment between $(0,0)$ and (d_x, d_y), or on the segment joining $(0, d_y)$ to $(d_x, 0)$. Let us first deal with the former case. Let $g \geq 0$ be the g.c.d. of d_x and d_y, and let u and v be the Bézout coefficients so that $g = u d_x + v d_y$ holds with $|u| \leq d_y$ and $|v| \leq d_x$. We refer the reader to [GG03, Lemma 3.12] for instance for these classical facts. We take U to be the linear map whose matrix is

$$\begin{pmatrix} u & v \\ -d_y/g & d_x/g \end{pmatrix}.$$

Since $\operatorname{Int}\mathcal{S} = \{(id_x/g, id_y/g) | i \in \{0, \ldots, g\}\}$ we have that $\pi = g + 1$ and that $U(\operatorname{Int}\mathcal{S})$ is the segment joining $(0, 0)$ to $(g, 0)$. It follows that $U(\mathcal{S})$ has dense size exactly π.

The latter case, where $\mathcal{S}$ is on the segment joining $(0, d_y)$ to $(d_x, 0)$, is similar, taking

$$U : (i, j) \mapsto \begin{pmatrix} -u & v \\ d_y/g & d_x/g \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix} + \begin{pmatrix} ud_x \\ -d_xd_y/g \end{pmatrix}.$$

By [GG03, Theorem 3.13] the computation of g, u, and v can be done with $O(\log^2 \delta)$ bit-operations using the naive version of the Euclidean algorithm. Then applying U to all the points of $\mathcal{S}$ takes $O\left(\sigma \log^2 \delta\right)$ bit-operations by multiplying naively the integers in quadratic time [GG03, Chapter 2, Section 3]. □

We remark that the value of η does not intervene in this degenerate case.

2.5. **Reduction algorithm.** Until the end of this section we assume that $\mathcal{S}$ is a nondegenerate finite set of points. The following algorithm computes $U \in \operatorname{Aff}\left(\mathbb{Z}^2\right)$ such that $U(\mathcal{S})$ is η-reduced.

Algorithm 1. Support reduction.

```
    Input: a nondegenerate normalized finite subset $\mathcal{S}$ of $\mathbb{N}^2$ of cardinality $\sigma$, and
        a real number $\eta$ in $\left[0, \frac{3}{4}\right)$.
    Output: $U \in \operatorname{Aff}(\mathbb{Z}^2)$, such that $U(\mathcal{S})$ is $\eta$-reduced.
  Compute $(d_x, d_y)$ for $\mathcal{S}$, as defined in Section 2.1.
  Initialize $U$ with the identity.
  Repeat
  (1) If $d_x < d_y$ then
          $\mathcal{S} := \mu(\mathcal{S})$
          $U := \mu \circ U$
          Swap $d_x$ and $d_y$.
  (2) Compute $b$, $d$, $f$, $h$ for $\mathcal{S}$, as defined in Section 2.1.
  (3) If $b + f > (1 + \eta)\, d_y$ then
          $\mathcal{S} := \tau_{d_y - f} \circ \lambda(\mathcal{S})$
          $U := \tau_{d_y - f} \circ \lambda \circ U$
          $d_x := d_x + d_y - b - f$
      else if $d + h > (1 + \eta)\, d_y$ then
          $\mathcal{S} := \tau_{-h} \circ \lambda^{-1}(\mathcal{S})$
          $U := \tau_{-h} \circ \lambda^{-1} \circ U$
          $d_x := d_x + d_y - d - h$
      else return $U$.
```

Proposition 2.3. *Algorithm 1 is correct. For any nondegenerate normalized finite subset $\mathcal{S}$ of $\mathbb{N}^2$ with bounding rectangle $[0, d_x] \times [0, d_y]$, Algorithm 1 performs at most $O(\max(d_x, d_y))$ steps in the main "Repeat" loop.*

Proof. After each reduction step in the main loop, either d_x and d_y are swapped, or d_x decreases by at least 1 and d_y is left unchanged. Therefore the number of steps is bounded by $O(\max(d_x, d_y))$. Since $\mathcal{S}$ remains normalized all along the process, the algorithm always terminates with $\mathcal{S}$ being η-reduced. □

Example 2.4. Assume $\eta = 0$ and let $F = 1 + xy + x^5y^2$, whose support $\mathcal{S}$ is $\{(0,0),(1,1),(5,2)\}$, as drawn in Figure 5 below. After the first step of the algorithm, where λ is applied, $\mathcal{S}$ becomes as in the left part of Figure 6. In the second step, λ is applied once more and makes $\mathcal{S}$ reduced, as shown in the right part of Figure 6. In the end, the algorithm returns $U = \tau_1 \circ \lambda^2$, so that we have $U(F) = x + y + x^2y^2$. The bounding rectangle of $U(F)$ corresponds to $d_x = d_y = 2$, while its bounding octagon $\mathcal{O}$ is defined by $b = f = h = 1$, and $d = 0$.

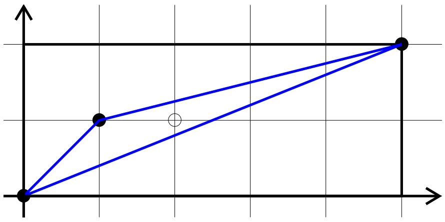

Figure 5. Input set $\mathcal{S} = \{(0,0),(1,1),(5,2)\}$.

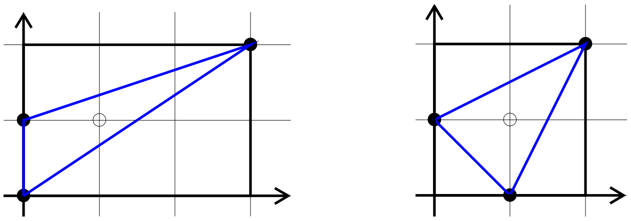

Figure 6. $\mathcal{S}$ after one, and then two reduction steps.

Example 2.5. Let $\mathcal{S}$ be $\{(0,0),(1,1),(5,2)\}$ as in Example 2.4 and Figure 5, and assume $\eta = 1/2$. Since $b + f = 4$ and $(1+\eta)\,d_y = 3$, the input set $\mathcal{S}$ can be reduced by applying λ to obtain the same set as in the left part of Figure 6. However, after this reduction, we have $b + f = 3$ which is not strictly greater than $(1+\eta)\,d_y = 3$. We thus see with this example that the reduction process stops earlier with $\eta = 1/2$ than with $\eta = 0$.

2.6. **Bit-cost analysis.** The main difficulty in analyzing the bit-cost of Algorithm 1 resides in bounding the size of the entries of the map U. This is the purpose of the following lemma:

Lemma 2.6. *Let $\mathcal{S}$ be a nondegenerate normalized finite subset $\mathcal{S}$ of $\mathbb{N}^2$ with bounding rectangle $[0,d_x] \times [0,d_y]$, and let U be an affine map as in* (1.1) *that sends $\mathcal{S}$ to a normalized set $\tilde{\mathcal{S}}$ with bounding rectangle $[0,\tilde{d}_x] \times [0,\tilde{d}_y]$. Then we have:*

- $|\alpha| \le 2\tilde{d}_x d_y$, $|\beta| \le 2d_x\tilde{d}_x$, $|\alpha'| \le 2d_y\tilde{d}_y$, *and* $|\beta'| \le 2d_x\tilde{d}_y$,
- $|\gamma| \le 4d_xd_y\tilde{d}_x$ *and* $|\gamma'| \le 4d_xd_y\tilde{d}_y$.

Proof. Since $\mathcal{S}$ is nondegenerate, then it contains at least three points $A = (x_A, y_A)$, $B = (x_B, y_B)$, and $C = (x_C, y_C)$ that are not aligned. Computing the images of A,

B, and C by the linear part $L = \begin{pmatrix} \alpha & \beta \\ \alpha' & \beta' \end{pmatrix}$ of U leads to:

$$\begin{cases} |\alpha(x_B - x_A) + \beta(y_B - y_A)| \leq \tilde{d}_x \\ |\alpha(x_C - x_A) + \beta(y_C - y_A)| \leq \tilde{d}_x. \end{cases}.$$

It follows that

$$\begin{cases} |\alpha(x_B - x_A)(x_C - x_A) + \beta(y_B - y_A)(x_C - x_A)| \leq d_x\tilde{d}_x \\ |\alpha(x_C - x_A)(x_B - x_A) + \beta(y_C - y_A)(x_B - x_A)| \leq d_x\tilde{d}_x, \end{cases}$$

whence

$$|\beta||(y_B - y_A)(x_C - x_A) - (y_C - y_A)(x_B - x_A)| \leq 2d_x\tilde{d}_x.$$

Since $|(y_B - y_A)(x_C - x_A) - (y_C - y_A)(x_B - x_A)|$ is a nonzero integer, we deduce that $|\beta| \leq 2d_x\tilde{d}_x$. The bounds for α, α' and β' can be obtained *mutatis mutandis*.

Since points of the image of $\mathcal{S}$ by L have abscissae (resp. ordinates) with absolute values at most $4d_xd_y\tilde{d}_x$ (resp. $4d_xd_y\tilde{d}_y$), the absolute value of γ (resp. γ') is at most $4d_xd_y\tilde{d}_x$ (resp. $4d_xd_y\tilde{d}_y$). □

Proposition 2.7. *For any nondegenerate normalized finite subset $\mathcal{S}$ of $\mathbb{N}^2$ of cardinality σ, with bounding rectangle $[0, d_x] \times [0, d_y]$, and dense size $\delta = (d_x+1)(d_y+1)$, Algorithm 1 takes $O(\sigma \max(d_x, d_y) \log \delta)$ bit-operations.*

Proof. Since the maximum of d_x and d_y never increases during the main loop, the bit-size of the points in $\mathcal{S}$ remains in $O(\log \delta)$. Therefore Lemma 2.6 implies that all the entries in U also remain bounded by $O(\log \delta)$ at every step. Each reduction step thus takes $O(\sigma \log \delta)$ bit-operations. The conclusion follows from Proposition 2.3. □

3. Dense size of reduced sets

Let $\mathcal{S}$ be a finite subset of $\mathbb{Z}^2$. In this section, we carry on using the notation of Section 2.1, and we further write $\operatorname{Vol} \mathcal{S}$ for the *volume of the convex hull* of $\mathcal{S}$. In the next paragraphs, we show that $\operatorname{Vol} \mathcal{S}$ cannot be too small compared to the volume $\operatorname{Vol} \mathcal{R}$ of the bounding rectangle $\mathcal{R}$ of $\mathcal{S}$, whenever $\mathcal{S}$ is reduced. In the second subsection, we deduce similar bounds in terms of discrete sizes while taking care of the degenerate cases.

3.1. Continuous bound.

Recall that η is a real constant in $[0, 3/4)$. The following theorem guarantees that the volume spanned by an η-reduced set of points can be uniformly controlled in terms of the volume of its bounding rectangle:

Theorem 3.1. *If $\mathcal{S}$ is an η-reduced set of points, then $\operatorname{Vol} \mathcal{S} \geq \frac{3-4\eta}{8} \operatorname{Vol} \mathcal{R}$, where $\mathcal{R}$ is the bounding rectangle of $\mathcal{S}$.*

Proof. In Lemma 3.2 below, we shall show that the volume of $\mathcal{S}$ is larger or equal to the volume of at least one of the following polygons:

$$\mathcal{Q}_1 = ACEG,$$
$$\mathcal{Q}_2 = BDFH,$$

$$\begin{aligned}
\mathcal{P}_1 &= ABDEG, & \mathcal{P}_2 &= BCEGH,\\
\mathcal{P}_3 &= BCEFH, & \mathcal{P}_4 &= BDEGH,\\
\mathcal{P}_5 &= ABDFG, & \mathcal{P}_6 &= ACDFH,\\
\mathcal{P}_7 &= ACEFH, & \mathcal{P}_8 &= ACDFG.
\end{aligned}$$

Then, Lemma 3.3 asserts that $\operatorname{Vol}\mathcal{Q}_i \geq \frac{1-\eta}{2}\operatorname{Vol}\mathcal{R}$, for all $i \in \{1,2\}$. And finally, for the eight pentagons, the combination of Lemmas 3.4 and 3.6 below provides us with $\operatorname{Vol}\mathcal{P}_i \geq \frac{3-4\eta}{8}\operatorname{Vol}\mathcal{R}$, for all $i \in \{1,\dots,8\}$. □

Lemma 3.2. *Let $\mathcal{S}$ be a normalized finite set of points (not necessarily η-reduced). Then at least one of the polygons $\mathcal{Q}_1$, $\mathcal{Q}_2$, $\mathcal{P}_1$,...,$\mathcal{P}_8$ defined above has at most* $\operatorname{Vol}\mathcal{S}$.

Proof. From the definitions of the bounding rectangle, and of b, d, f, h, there exist eight points I, J, K, L, M, N, O and P in $\mathcal{S}$ such that $I \in [AB]$, $J \in [BC], \dots, P \in [AH]$, as drawn on the following figure (note that some of these points may coincide in particular degenerate cases):

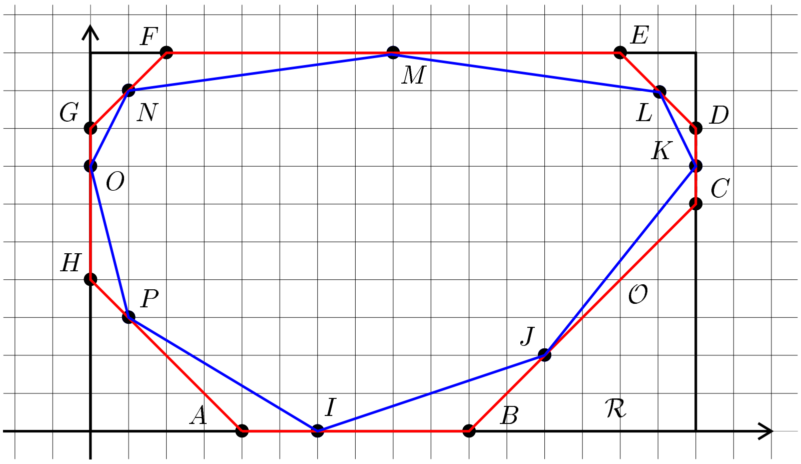

Figure 7. Points of $\mathcal{S}$ lying on the bounding octagon $\mathcal{O}$.

Since $\operatorname{Vol}\mathcal{S}$ is the volume of the convex hull spanned by $\mathcal{S}$, it is already clear that

$$\operatorname{Vol}(IJKLMNOP) \leq \operatorname{Vol}\mathcal{S}.$$

By considering the subdivision of $IJKLMNOP$ into the triangle IJP and the polygon $JKLMNOP$, we see that $\operatorname{Vol}(AJP) \leq \operatorname{Vol}(IJP)$ or $\operatorname{Vol}(BJP) \leq \operatorname{Vol}(IJP)$, according to the slope of (PJ) being positive or not.
It follows that $\operatorname{Vol}(AJKLMNOP) \leq \operatorname{Vol}\mathcal{S}$ or $\operatorname{Vol}(BJKLMNOP) \leq \operatorname{Vol}\mathcal{S}$. In other words, moving I on its supporting segment $[A,B]$ makes $\operatorname{Vol}(IJKLMNOP)$ either decrease or increase. Doing so with K, M and O, and then with some points among J, L, N and P, so that $\operatorname{Vol}(IJKLMNOP)$ decreases, we are led to distinguish the following cases:

- If I, K, M and O all move clockwise, that is I moves to A, K moves to C, M moves to E and O moves to G, then we get the polygon $AJCLENGP$ whose volume is at least $\operatorname{Vol}\mathcal{Q}_1$.

- If I, K, M and O all move counterclockwise, then we get the polygon $BJDLFNHP$ whose volume is at least $\operatorname{Vol}\mathcal{Q}_2$.
- Otherwise two consecutive points among the cycle I, K, M, and O move into opposite directions. We now remark that the symmetries $i \mapsto d_x - i$, $j \mapsto d_y - j$ and $(i,j) \mapsto (j,i)$ preserve the problem, the volumes, exchange the roles of $\mathcal{Q}_1$ and $\mathcal{Q}_2$, and globally preserve the set of the eight pentagons $\mathcal{P}_1, \ldots, \mathcal{P}_8$. We can thus restrict ourselves to considering for instance the case for when I moves to B and K moves to C, and examine the following subcases:
  - M moves to E and O to H. If N moves to F, then we get the polygon $BCLEFHP$, that has volume at least $\operatorname{Vol}\mathcal{P}_3$. Otherwise, if N moves to G, then we get the polygon $BCLEGHP$, which has volume at least $\operatorname{Vol}\mathcal{P}_2$.
  - M moves to E and O to G. If P moves to H, then we get the polygon $BCLENGH$, that has volume at least $\operatorname{Vol}\mathcal{P}_2$. Otherwise, if P moves to A, then we get the polygon $BCLENGA$, which has volume at least $\operatorname{Vol}\mathcal{Q}_1$.
  - M moves to F and O to H. If L moves to D then we get the polygon $BCDFNHP$, that has volume at least $\operatorname{Vol}\mathcal{Q}_2$. Otherwise, if L moves to E, then we get the polygon $BCEFNHP$, which has volume at least $\operatorname{Vol}\mathcal{P}_3$.
  - M moves to F and O to G. Let us assume that P moves to A. Then if L moves to D, then we get the polygon $BCDFNGA$, which has volume at least $\operatorname{Vol}\mathcal{P}_5$. Otherwise, if L moves to E then we get the polygon $BCEFNGA$, that has volume at least $\operatorname{Vol}\mathcal{Q}_1$. The symmetries then handle the situation of P moving to H instead of A. □

Lemma 3.3. *If $\mathcal{S}$ is an η-reduced set of points, then*

$$\operatorname{Vol}\mathcal{Q}_i \geq \frac{1-\eta}{2}\operatorname{Vol}\mathcal{R}, \text{ for } i \in \{1,2\}.$$

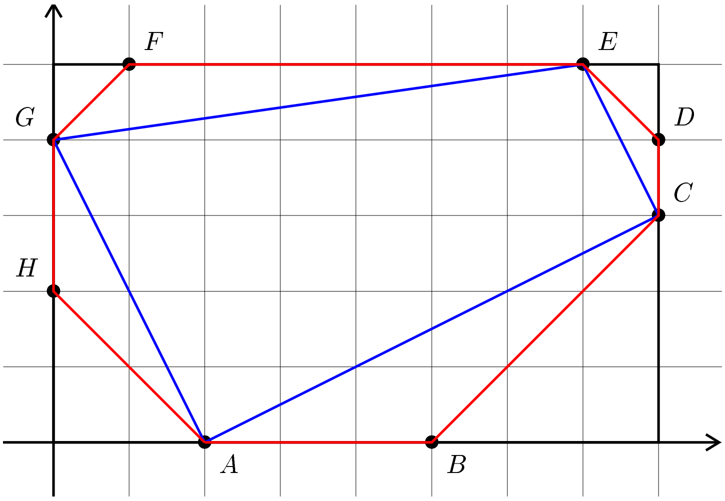

FIGURE 8. Quadrangle $\mathcal{Q}_1$ in octagon $\mathcal{O}$ and rectangle $\mathcal{R}$.

Proof. Since the roles of $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are interchanged by the symmetry $i \mapsto d_x - i$, it suffices to prove the lemma for $\mathcal{Q}_1$ only. We compute the volume of $\mathcal{Q}_1$ as the difference between the volume $d_x d_y$ of the bounding rectangle and the volume of the four triangles outside of $\mathcal{Q}_1$:

$$\begin{aligned}\operatorname{Vol}\mathcal{Q}_1 &= d_x d_y - (d_x - h)b/2 - (d_y - b)d/2 - (d_x - d)f/2 - (d_y - f)h/2 \\ &= \frac{1}{2}(d_y - b - f)(d_x - d - h) + \frac{1}{2} d_x d_y. \qquad (3.1)\end{aligned}$$

Since $\mathcal{S}$ is η-reduced, we have $(1+\eta)\, d_y - b - f \geq 0$, $d_x - d - h \geq 0$, thus $d_y - b - f \geq -\eta d_y$. This yields $\operatorname{Vol}\mathcal{Q}_1 \geq \frac{1}{2} d_x d_y - \frac{\eta}{2} d_x d_y = \frac{1-\eta}{2}\operatorname{Vol}\mathcal{R}$. □

Lemma 3.4. *If $\mathcal{S}$ is an η-reduced set of points, then*

$$\operatorname{Vol}\mathcal{P}_i \geq \frac{3-4\eta}{8}\operatorname{Vol}\mathcal{R}, \textit{ for } i \in \{1,3,5,7\}.$$

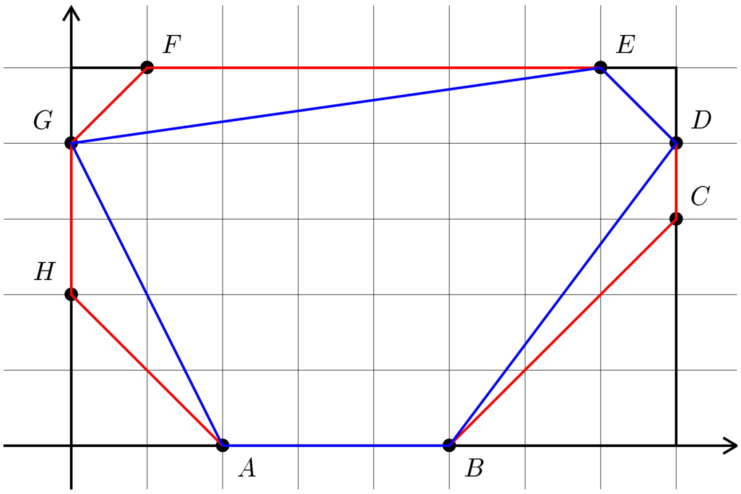

Figure 9. Pentagon $\mathcal{P}_1$ in octagon $\mathcal{O}$ and rectangle $\mathcal{R}$.

Proof. Thanks to the symmetries, it suffices to prove the lemma for $\mathcal{P}_1$. The volume of $\mathcal{P}_1$ is computed as the difference of the volume of $\mathcal{R}$ with those of the four triangles outside of $\mathcal{P}_1$:

$$\operatorname{Vol}\mathcal{P}_1 = \operatorname{Vol}\mathcal{R} - \frac{1}{2}\big(b\,(d_y - d) + d^2 + f\,(d_x - d) + h\,(d_y - f)\big).$$

From (3.1) we deduce that:

$$\operatorname{Vol}\mathcal{P}_1 - \operatorname{Vol}\mathcal{Q}_1 = \frac{1}{2}\left(b\,(d_x - d_y - h) + d\,(d_y - d)\right).$$

Then, from

$$4b\,(d_x - d_y - h) + d_y^2 = 4b(d_x - b - h) + (2b - d_y)^2,$$

and $d_x - b - h \geq 0$, it follows that $4b\,(d_x - d_y - h) + d_y^2 \geq 0$, and that $\operatorname{Vol}\mathcal{P}_1 - \operatorname{Vol}\mathcal{Q}_1 \geq -\frac{1}{8}d_y^2$. The conclusion comes from Lemma 3.3:

$$\operatorname{Vol}\mathcal{P}_1 \geq \frac{1-\eta}{2} d_x d_y - \frac{1}{8} d_y^2 \geq \frac{3-4\eta}{8}\operatorname{Vol}\mathcal{R}. \qquad \square$$

Remark 3.5. For $\eta = 0$, the inequality of Lemma 3.4 turns out to be sharp. For instance with $\mathcal{S} = \{(d_x/2, 0), (0, d_x/2), (d_x, d_x)\}$ we have $b = f = h = \frac{1}{2} d_x$ and $d = 0$. Pentagon $\mathcal{P}_1$, as drawn on the following figure, has volume $\frac{3}{8} d_x^2$.

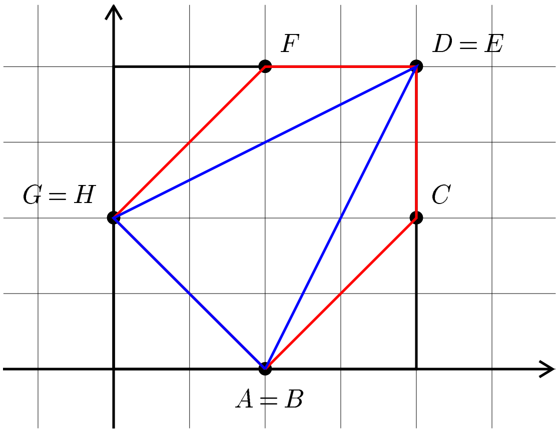

FIGURE 10. Minimal pentagon $\mathcal{P}_1$ with $d_x = d_y$.

Lemma 3.6. *If $\mathcal{S}$ is an η-reduced set of points, then*

$$\operatorname{Vol} \mathcal{P}_i \geq \frac{3 - 4\eta}{8} \operatorname{Vol} \mathcal{R}, \quad \textit{for } i \in \{2, 4, 6, 8\}.$$

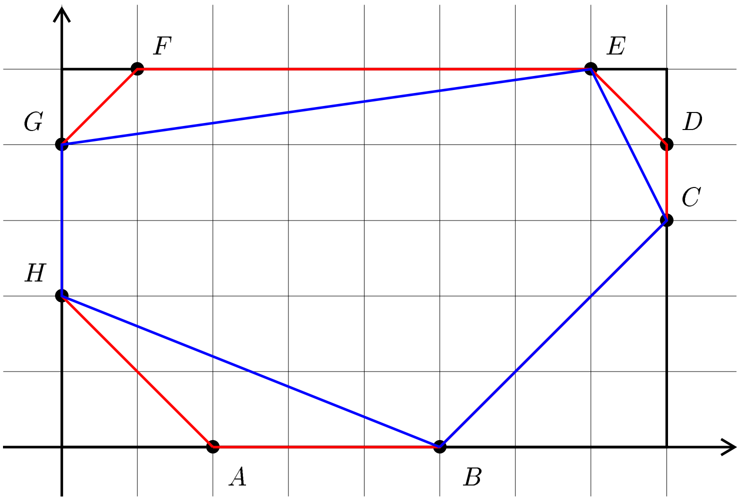

FIGURE 11. Pentagon $\mathcal{P}_2$ in octagon $\mathcal{O}$ and rectangle $\mathcal{R}$.

Proof. Thanks to the symmetries, it suffices to prove the lemma for $\mathcal{P}_2$. Specifically we shall prove that the following quantity is nonnegative:

$$\begin{aligned}\theta(b,d,f,h) &= 8\operatorname{Vol}\mathcal{P}_2 - (3-4\eta)\operatorname{Vol}\mathcal{R}\\ &= (5+4\eta)\,d_x d_y + 4h\,(b-d_x) - 4b^2 + 4d\,(b-d_y) + 4f\,(d-d_x)\\ &= (1+4\eta)\,d_x d_y + 4\,(d_x-d)\,(d_y-f-h) - 4\,(b-d)\,(b-h).\end{aligned}$$

Since $f+h \le d_y$, we have that $(1+4\eta)\,d_x d_y + 4\,(d_x-d)\,(d_y-f-h) \ge 0$. Therefore, if $h \ge b \ge d$ or $d \ge b \ge h$, then the lemma is proved.

Otherwise, if $b \le d$ and $b \le h$, $|b-d|\,|b-h|$ is maximal for $b=0$ and, for $d = h = (1+\eta)\,d_y/2$, since $d+h \le (1+\eta)\,d_y$. It follows that $-4\,(b-d)\,(b-h) \ge -(1+\eta)^2\,d_y^2$. From $d_x \ge d_y$ and $\eta \in [0,3/4)$ we deduce that $(1+4\eta)\,d_x \ge (1+\eta)^2\,d_y$, and that $\theta(b,d,f,h) \ge 0$.

It remains to study the case for when $b \ge d$ and $b \ge h$. Using $d_x - d \ge b - d$, we obtain:

$$\theta(b,d,f,h) \ge (1+4\eta)\,d_x d_y + 4\,(b-d)\,(d_y-b-f).$$

Then applying $b+f \le (1+\eta)d_y$ leads to:

$$\begin{aligned}\theta(b,d,f,h) &\ge (1+4\eta)\,d_x d_y - 4\,(b-d)\,\eta d_y\\ &\ge (1+4\eta)\,d_x d_y - 4\eta d_x d_y \ge 0,\end{aligned}$$

which concludes the proof. □

3.2. **Discrete bound.** For our algorithmic purposes, we need to control the number of integral points in the convex hull, instead of its volume.

Proposition 3.7. *If $\mathcal{S}$ is an η-reduced subset of $\mathbb{N}^2$ of convex size π and with bounding rectangle $\mathcal{R} = [0,d_x] \times [0,d_y]$, then the following inequalities hold:*

$$\frac{3-4\eta}{18}(d_x+1)(d_y+1) \le \pi \le (d_x+1)(d_y+1).$$

Proof. As $\mathcal{R}$ contains $\mathcal{S}$, the convex size π is always at most $(d_x+1)(d_y+1)$. If $\mathcal{S}$ is degenerate, then $d_y = 0$ and $\pi = d_x + 1$, so that the proposition is correct. Let us now assume that $\mathcal{S}$ is nondegenerate. We decompose $\operatorname{Int}\mathcal{S}$ into $\operatorname{Int}_b\mathcal{S} \cup \operatorname{Int}_i\mathcal{S}$, where $\operatorname{Int}_b\mathcal{S}$ are the points lying upon the boundary of the Newton polygon of $\mathcal{S}$, while $\operatorname{Int}_i\mathcal{S}$ are the other ones strictly inside. Pick's Theorem (see [Cox69, Chapter 13, Proposition 51] or [GS93]) relates $\operatorname{Vol}\mathcal{S}$ to $|\operatorname{Int}_b\mathcal{S}|$ and $|\operatorname{Int}_i\mathcal{S}|$, as follows:

$$\operatorname{Vol}\mathcal{S} = \frac{1}{2}|\operatorname{Int}_b\mathcal{S}| + |\operatorname{Int}_i\mathcal{S}| - 1.$$

It follows that $\pi \ge \operatorname{Vol}\mathcal{S}$, and that $\pi \ge \frac{3-4\eta}{8} d_x d_y$ by Theorem 3.1.

Whenever $d_y = 1$, we have $b+d+f+h \in \{0,1,2\}$. If $b+h = 1$ and $d+f = 1$, then from $b+f \le 1$ and $d+h \le 1$, we can deduce that $f = h$ which implies $f = h = 0$ because $f+h \le 1$. Therefore, $b = d = 1$, which is impossible since $b+d \le 1$. Finally, we must have $b+h = 0$ or $d+f = 0$, hence $\pi \ge d_x + 1$.

If $d_y \ge 2$, the conclusion follows from $d_x \ge 2(d_x+1)/3$ and $d_y \ge 2(d_y+1)/3$. □

Remark 3.8. In the case for when $\eta = 0$, if α is such that the inequality $\alpha|\operatorname{Int}\mathcal{R}| \le |\operatorname{Int}\mathcal{S}|$ holds for every reduced finite subset $\mathcal{S}$ in $\mathbb{Z}^2$, with $\operatorname{Vol}\mathcal{S} > 0$, then necessarily we have that $\alpha \le 3/8$. In fact it suffices to consider the family $\mathcal{S}_n =$

$\{(n/2,0),(0,n/2),(n,n)\}$ for n even. We have $|\operatorname{Int}\mathcal{S}_2| = 4$ and $|\operatorname{Int}\mathcal{S}_{n+2}| = |\operatorname{Int}\mathcal{S}_n| + 3\left(\frac{n}{2}+1\right)$, and deduce that

$$\frac{|\operatorname{Int}\mathcal{S}_n|}{|\operatorname{Int}\mathcal{R}_n|} = \frac{3n\,(n+2)+8}{8\,(n+1)^2}$$

is decreasing and converges to 3/8. In general, the constant $\frac{3-4\eta}{18}$ thus may be rather pessimistic for large $\mathcal{S}$.

4. Faster reduction algorithm

The last ingredient now missing to prove Theorem 1.2 is a reduction algorithm with a number of reduction steps that grows only with the logarithm of the dense size. This is the goal of this section.

4.1. **Dichotomic approach.** This section is dedicated to a fast variant of Algorithm 1. We do not compute exactly the same output, however. Roughly speaking, the main idea is to determine quickly how many times λ or λ^{-1} can be applied before two consecutive swaps.

Let $\mathcal{S}$ be a normalized finite subset of $\mathbb{N}^2$ with bounding rectangle $[0,d_x]\times[0,d_y]$, and let q be a positive integer. The points (i,j) in $\mathcal{S}$ that are sent to the far left of $\lambda^q(\mathcal{S})$ are such that $i-qj$ is minimal. Analogously, those that are sent to the far right of $\lambda^q(\mathcal{S})$ are such that $i-qj$ is maximal. This motivates the introduction of b_q, d_q, f_q, and h_q as

- $b_q = d_x - \max_{(i,j)\in\mathcal{S}}(i-qj)$,
- $d_q = d_x + qd_y - \max_{(i,j)\in\mathcal{S}}(i+qj)$,
- $f_q = qd_y + \min_{(i,j)\in\mathcal{S}}(i-qj)$,
- $h_q = \min_{(i,j)\in\mathcal{S}}(i+qj)$.

For $q=1$, these definitions coincide with those of b, d, f, and h of Section 2.1. The following inequalities hold:

$$\begin{aligned} b_q + d_q &\le qd_y, & \qquad b_q + h_q &\le d_x,\\ f_q + h_q &\le qd_y, & \qquad d_q + f_q &\le d_x. \end{aligned}$$

The height of the bounding rectangle of $\lambda^q(\mathcal{S})$ is still d_y, while the horizontal length becomes $d_x + qd_y - b_q - f_q$. In the same manner, the horizontal length of the bounding rectangle of $\lambda^{-q}(\mathcal{S})$ becomes $d_x + qd_y - d_q - h_q$.

From now on, the reduction factor η is supposed to be positive, that is in $(0,3/4)$. We write $\lfloor a\rfloor$ for the integer part of a ($\lfloor a\rfloor \le a < \lfloor a\rfloor + 1$), and $\log_2 a$ for the logarithm of a in base 2. The fast algorithm we propose summarizes as follows:

Algorithm 2. Dichotomic support reduction.

Input: a nondegenerate normalized finite subset $\mathcal{S}$ of $\mathbb{N}^2$ of cardinality σ, and a real number η in $\left(0,\frac{3}{4}\right)$.

Output: $U \in \operatorname{Aff}(\mathbb{Z}^2)$, such that $U(\mathcal{S})$ is η-reduced.

```
Compute $(d_x, d_y)$ for $\mathcal{S}$, as defined in Section 2.1.
Initialize $U$ with the identity.
Initialize $m$ with $\lfloor \log_2(d_x/(\eta d_y)) \rfloor$.
Repeat
 (1) If $d_x < d_y$ then
```

```
            $\mathcal{S} := \mu(\mathcal{S})$
            $U := \mu \circ U$
            Swap $d_x$ and $d_y$
            $m := \lfloor \log_2(d_x/(\eta d_y)) \rfloor$.
    (2) If $m < 0$ then return $U$.
    (3) Compute $b_{2^m}$, $d_{2^m}$, $f_{2^m}$, $h_{2^m}$ for $\mathcal{S}$ as defined above.
    (4) If $b_{2^m} + f_{2^m} > 2^m(1+\eta)\, d_y$ then
            $\mathcal{S} := \tau_{2^m d_y - f_{2^m}} \circ \lambda^{2^m}(\mathcal{S})$
            $U := \tau_{2^m d_y - f_{2^m}} \circ \lambda^{2^m} \circ U$
            $d_x := d_x + 2^m d_y - b_{2^m} - f_{2^m}$
        else if $d_{2^m} + h_{2^m} > 2^m(1+\eta)\, d_y$ then
            $\mathcal{S} := \tau_{-h_{2^m}} \circ \lambda^{-2^m}(\mathcal{S})$
            $U := \tau_{-h_{2^m}} \circ \lambda^{-2^m} \circ U$
            $d_x := d_x + 2^m d_y - d_{2^m} - h_{2^m}$.
    (5) $m := m - 1$.
```

Proposition 4.1. *Assume that $\eta > 0$. For any nondegenerate normalized finite subset $\mathcal{S}$ of $\mathbb{N}^2$, of cardinality σ and dense size δ, Algorithm 2 is correct and runs in $O\left(\sigma \log^2 \delta\right)$ bit-operations.*

Proof. Let us consider that the bounding rectangle of $\mathcal{S}$ is $[0, d_x] \times [0, d_y]$ at input. Without loss of generality, we can assume that $d_x \geq d_y$ holds in order to simplify the proof. Then we let $\ell_0 = d_x$ and $\ell_1 = d_y$, and define the sequence $(\mathcal{S}_i)_i$ with $\mathcal{S}_0 = \mathcal{S}$ and $\mathcal{S}_i$ is the current value of the set just after the ith swap, that is at the end of step 1. We write r for the total number of swaps performed during the execution of the algorithm, we let ℓ_i be the largest abscissa in $\mathcal{S}_i$, and m_i be $\lfloor \log_2(\ell_i/(\eta \ell_{i+1})) \rfloor$. By convention, ℓ_{r+1} is the largest ordinate in $\mathcal{S}_r$.

For when $i+2 \leq r$ holds, we have that $\ell_{i+2} \leq \ell_i - \eta \ell_{i+1}$. By descending induction, starting with $\ell_r \geq 1$ and $\ell_{r-1} \geq 1$, we shall prove that $\ell_i \geq \varphi^{r-i-1}$, where φ is the positive root $\frac{\eta+\sqrt{4+\eta^2}}{2} > 1$ of the characteristic equation $x^2 - \eta x - 1 = 0$. Since this is true for $i = r$ and $i = r-1$, and since $\ell_i \geq \eta \ell_{i+1} + \ell_{i+2} \geq \eta \varphi^{r-i-2} + \varphi^{r-i-3} = \varphi^{r-i-1}$, we deduce that $d_x = \ell_0 \geq \varphi^{r-1}$. The number of swaps r thus drops to $O(\log d_x)$.

By Lemma 2.6, since all the $\mathcal{S}_i$ are normalized, each reduction step amounts to $O(\sigma \log \delta)$ bit-operations. On the other hand the total number of steps is $\sum_{i=0}^r m_i$,

$$\sum_{i=0}^{r} m_i \in O\left(\sum_{i=0}^{r} \log_2\left(\frac{\ell_i}{\ell_{i+1}}\right) + r \log_2 \frac{1}{\eta}\right) \in O(\log \delta),$$

which concludes the cost analysis.

We shall prove that when the algorithm stops, the final value of $\mathcal{S}$ is η-reduced. We now focus on what happens just after the last swap. In short, we let M be m_r and $\mathcal{T}_{M+1}$ be $\mathcal{S}_r$. We denote by $\mathcal{T}_m$ the current value of $\mathcal{S}$ just before entering step 6, where m being the corresponding current value of m. Therefore $\mathcal{T}_0$ corresponds to the output of the algorithm and we want to prove that it is η-reduced. If $\mathcal{T}_0 = \mathcal{T}_1$ then we are done.

If $\mathcal{A}$ is a subset of points, then we write $\ell_x(\mathcal{A})$ for the horizontal length of the bounding rectangle of $\mathcal{A}$. Let us now assume that $\mathcal{T}_0$ is the normalization of $\lambda(\mathcal{T}_1)$. In this case, of course, λ^{-1} does not reduce $\mathcal{T}_0$. Let us prove that λ does not reduce $\mathcal{T}_0$ either. If $\mathcal{T}_m$ were the normalization of $\lambda^{2^m}(\mathcal{T}_{m+1})$ for all m in $\{0, \ldots, M\}$, then we would deduce that

$$\begin{aligned}\ell_x(\mathcal{T}_0) \le \ell_x(\mathcal{T}_{M+1}) - \sum_{m=0}^{M} 2^m \eta \ell_{r+1} &= \ell_r - (2^{M+1}-1)\eta\ell_{r+1} \\ &< \ell_r - (\ell_r/(\eta\ell_{r+1}) - 1)\eta\ell_{r+1} = \eta\ell_{r+1},\end{aligned}$$

which is impossible. Therefore there exists a largest integer $\mu \in \{0,\dots,M\}$ such that for all m in $\{0,\dots,\mu-1\}$, $\mathcal{T}_m$ is the normalization of $\lambda^{2^m}(\mathcal{T}_{m+1})$. This yields that $\mathcal{T}_0$ is the normalization of $\lambda^{2^\mu-1}(\mathcal{T}_\mu)$ and also that

$$\ell_x(\mathcal{T}_0) < \ell_x(\mathcal{T}_\mu) - (2^\mu - 1)\,\eta\ell_{r+1}. \tag{4.1}$$

One of the following two cases arises:

- If $\mathcal{T}_\mu = \mathcal{T}_{\mu+1}$ then we have that
  $$\ell_x(\lambda(\mathcal{T}_0)) = \ell_x(\lambda^{2^\mu}(\mathcal{T}_\mu)) = \ell_x(\lambda^{2^\mu}(\mathcal{T}_{\mu+1})) \ge \ell_x(\mathcal{T}_{\mu+1}) - 2^\mu\eta\ell_{r+1}.$$
  Combined with (4.1) it follows that $\ell_x(\lambda(\mathcal{T}_0)) > \ell_x(\mathcal{T}_0) - \eta\ell_{r+1}$, and hence that $\mathcal{T}_0$ is η-reduced.
- Otherwise, if $\mathcal{T}_\mu$ is the normalization of $\lambda^{-2^\mu}(\mathcal{T}_{\mu+1})$, then we have that $\ell_x(\mathcal{T}_\mu) < \ell_x(\mathcal{T}_{\mu+1}) - 2^\mu\eta\ell_{r+1}$, so that
  $$\ell_x(\mathcal{T}_0) \le \ell_x(\mathcal{T}_{\mu+1}) - (2^{\mu+1}-1)\eta\ell_{r+1}.$$
  Since $\mathcal{T}_{\mu+1}$ is the normalization of $\lambda(\mathcal{T}_0)$, we deduce that
  $$\ell_x(\lambda(\mathcal{T}_0)) = \ell_x(\mathcal{T}_{\mu+1}) \ge \ell_x(\mathcal{T}_0) + (2^{\mu+1}-1)\eta\ell_{r+1},$$
  whence that $\mathcal{T}_0$ is η-reduced.

Finally the last case for when $\mathcal{T}_0$ is the normalization of $\lambda^{-1}(\mathcal{T}_1)$ can be treated in the same way. □

Remark 4.2. At each step of the algorithm, the current degree, say $\tilde{d}_y$, in y, is smaller than the initial d_y. Therefore whenever one chooses $\eta < \frac{1}{d_y}$ from the outset, the reduction is at least $\left\lceil \frac{\tilde{d}_y}{d_y} \right\rceil = 1$. Therefore the algorithms behaves the same as with $\eta = 0$ and Proposition 4.1 is still valid if $\eta = 0$. However, there is no uniform $\eta > 0$ such that, for every input, the behaviour of the algorithm is the same as with $\eta = 0$.

4.2. Proof of Theorem 1.2.

We prove a slightly more precise theorem than Theorem 1.2, which contains bounds on the bit-size of the entries of the affine map U:

Theorem 4.3. *For any normalized finite subset $\mathcal{S}$ of $\mathbb{Z}^2$, of cardinality σ, convex size π, bounding rectangle $[0,d_x]\times[0,d_y]$, and dense size $\delta = (d_x+1)(d_y+1)$, one can compute an invertible affine map $U \in \mathrm{Aff}(\mathbb{Z}^2)$ as in (1.1), with $O\left(\sigma \log^2 \delta\right)$ bit-operations, such that:*

- *$|\alpha|$, $|\beta|$, $|\alpha'|$, and $|\beta'|$ are at most $\max(2\max(d_x,d_y)^2, 1)$,*
- *$|\gamma|$ and $|\gamma'|$ are at most $4\max(d_x,d_y)^3$,*
- *$U(\mathcal{S})$ is normalized of dense size at most 9π.*

Proof. Proposition 2.2 already covers the degenerate case. In the nondegenerate situation, the theorem follows from Proposition 3.7 for the dense size of the output, from Proposition 4.1 with taking $\eta = 1/4$ for the bit-complexity, and from Lemma 2.6 for the size of the entries of U. □

4.3. **Timings.** We report on performances obtained with our implementation in Maple 14 for computing the irreducible factorization of the following polynomials in $\mathbb{Q}[x,y]$:

$$P_n = \left(x^{n+1} + \sum_{i=0}^{n} i x^i y^{n-i}\right)\left(y^{n+1} + \sum_{i=0}^{n}(n-i)x^i y^{n-i}\right) \times \left(x^{\lfloor \frac{n}{2} \rfloor -1} y^{\lfloor \frac{n}{2} \rfloor -1} + \sum_{i=0}^{n} x^i y^{n-i}\right).$$

The source code is available from
`http://www.lix.polytechnique.fr/~berthomieu/convex-dense.htm`.
In Table 1, we display timings, in seconds obtained using an Intel Xeon X5450 at 3.0 GHz running Linux. The first line contains the time spent in the direct call of the native function `factor`. The second line concerns the time spent in our Algorithm 1 with $\eta = 0$. The last line corresponds to calling `factor` on the reduced polynomial. Indeed, as an optimization, Algorithm 1 is run on the set of vertices of the convex hull of the support of the input polynomial. For a set of size σ, it is classical that the convex hull can be computed in time softly linear in σ: we refer the reader to [PS85, Chapter 3, Theorem 3.7] for instance.

n	8	16	32	64	128
dense factorization	0.04	0.25	2.3	48	1100
reduction	0.06	0.14	0.28	0.54	1.1
convex factorization	0.04	0.06	0.22	1.5	25

Table 1. Factorization of P_n, in seconds.

As expected, our reduction strategy leads to a significant speedup. In fact, with this family, notice that the dense size grows with n^2 while the convex size only grows with n. We have also tried Algorithm 2: the gains are not substantial since most of the time is spent in the factorization. Finally let us mention that one could investigate the design of a reduction algorithm featuring a dichotomy in the size of the exponents, in a way similar to the half-g.c.d. algorithm (see for instance [GG03, Chapter 11]). This would probably lead to a bit-complexity bound in $\tilde{O}(\sigma \log \delta)$. However, the practical impact would be minor as long as the sizes of the exponents are intended to fit into one machine word.

4.4. **Optimality of the reduction.** It is natural to ask if our algorithm computes the best transformation U of $\mathbb{Z}^2$, that maximizes the ratio of the volumes of $U(\mathcal{S})$ and $\mathcal{R}(U(\mathcal{S}))$, where $\mathcal{R}(U(\mathcal{S}))$ represents the bounding rectangle of $U(\mathcal{S})$.

First, let us mention that the transformations λ, μ and τ_1 used within our algorithm actually generate $\operatorname{Aff}(\mathbb{Z}^2)$. In fact it is classical that $\operatorname{SL}(\mathbb{Z}^2)$ is generated by λ and the rotation $\rho = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$ by the angle $\pi/2$ [Ser96, Chapter 7, Theorem 2]. Since ρ can be decomposed into $\rho = \mu\lambda\mu\lambda^{-1}\mu\lambda\mu$, and since $\det \mu = -1$, we deduce that λ and μ generate $\operatorname{GL}(\mathbb{Z}^2)$. However we will not prove that our algorithm returns the best $U \in \operatorname{Aff}(\mathbb{Z}^2)$ on all input. Roughly speaking, we will only prove that the bound $3/8$ of the ratio of the volumes at the end of our reduction algorithm is the best bound one can expect in general when $\eta = 0$. This bound is attained with the example of Figure 10. Specifically, we aim at proving there is no transformation U such that for all finite subset $\mathcal{S} \subset \mathbb{Z}^2$, the inequality $\operatorname{Vol} U(\mathcal{S}) \geq \alpha \operatorname{Vol} \mathcal{R}(U(\mathcal{S}))$ holds with $\alpha > 3/8$:

Proposition 4.4. *With the convention* $\frac{\operatorname{Vol} U(\mathcal{S})}{\operatorname{Vol} \mathcal{R}(U(\mathcal{S}))} = 1$ *whenever* $\operatorname{Vol} \mathcal{S} = 0$*, one has*

$$\inf_{\mathcal{S} \subset \mathbb{Z}^2, |\mathcal{S}| < \infty} \sup_{U \in \operatorname{Aff}(\mathbb{Z}^2)} \frac{\operatorname{Vol} U(\mathcal{S})}{\operatorname{Vol} \mathcal{R}(U(\mathcal{S}))} = \frac{3}{8},$$

where $\mathcal{R}(U(\mathcal{S}))$ *represents the bounding rectangle of* $U(\mathcal{S})$*.*

Proof. The degenerate case, that is when $\operatorname{Vol} \mathcal{S} = 0$, follows from Proposition 2.2, so that from now on, we can assume that $\operatorname{Vol} \mathcal{S} > 0$. By Theorem 3.1, there exists $U \in \operatorname{Aff}(\mathbb{Z}^2)$ such that $\operatorname{Vol} U(\mathcal{S}) \geq \frac{3}{8} \operatorname{Vol} \mathcal{R}(U(\mathcal{S}))$, whence

$$\inf_{\mathcal{S} \subset \mathbb{Z}^2, |\mathcal{S}| < \infty} \sup_{U \in \operatorname{Aff}(\mathbb{Z}^2)} \frac{\operatorname{Vol} U(\mathcal{S})}{\operatorname{Vol} \mathcal{R}(U(\mathcal{S}))} \geq \frac{3}{8}.$$

We shall show that $\sup_{U \in \operatorname{Aff}(\mathbb{Z}^2)} \frac{\operatorname{Vol} U(\mathcal{S})}{\operatorname{Vol} \mathcal{R}(U(\mathcal{S}))} = \frac{3}{8}$ holds for when $\mathcal{S} = \{(1,0), (0,1), (2,2)\}$, which will conclude the proof. For the rest of the proof, $\mathcal{S}$ represents the particular set of points $\{(1,0), (0,1), (2,2)\}$. As $\operatorname{Vol} U(\mathcal{S})$ is constant, and equals $3/2$ for all U, it suffices to show that, for any $U \in \operatorname{Aff}\left(\mathbb{Z}^2\right)$, $\operatorname{Vol} \mathcal{R}(U(\mathcal{S})) \geq 4$. As translating and swapping x and y do not change $\operatorname{Vol} \mathcal{R}(U(\mathcal{S}))$, we can assume that $U \in \operatorname{SL}\left(\mathbb{Z}^2\right)$. Let $\left(\begin{smallmatrix} \alpha & \beta \\ \alpha' & \beta' \end{smallmatrix}\right)$ be the matricial representation of U, with $\alpha\beta' - \beta\alpha' = 1$. Let ρ be the rotation by the angle $\pi/2$. As $\operatorname{Vol} \mathcal{R}(\rho(U(\mathcal{S}))) = \operatorname{Vol} \mathcal{R}(U(\mathcal{S}))$, one can apply ρ, or ρ^{-1}, once or twice so that we can further assume that $\alpha \geq 1$ and $\alpha' \geq 0$ hold.

If $\alpha' = 0$, then $\alpha\beta' = 1$ so that $\alpha = 1$ and $\beta' = 1$. Since the image of $(2,2)$ is $(2 + 2\beta, 2)$, the height of the bounding rectangle of $U(\mathcal{S})$ is 2, and $\operatorname{Vol} \mathcal{R}(U(\mathcal{S})) \geq 4$ as soon as the horizontal length of $\mathcal{R}(U(\mathcal{S}))$ is greater or equal to 2. In fact, this length is the maximum of $|\beta - 1|$, $|2\beta + 1|$ and $|\beta + 2|$. If $|\beta - 1| = 0$, then $\beta = 1$ and $2\beta + 1 = 3$. Otherwise, if $|\beta - 1| = 1$ then either $\beta = 0$ and $\beta + 2 = 2$, or $\beta = 2$ and $\beta + 2 = 4$. In this way we observe that, in all cases, the length is at least 2. We can now restrict to considering $\alpha' \geq 1$.

If $\beta = 0$ then $\alpha = 1$ and $\beta' = 1$. The horizontal length of $U(\mathcal{S})$ is 2 and its height is $2\alpha' + 1$. Therefore we have again that $\operatorname{Vol} \mathcal{R}(U(\mathcal{S})) \geq 4$. Similarly, when $\beta' = 0$, we have $\beta = -1$ and $\alpha' = 1$: the height of $U(\mathcal{S})$ is 2 and its horizontal length is at least $\alpha + 1 \geq 2$, which yields the same conclusion. Thus, we can now further restrict to considering the case that none of the coefficients of the matrix of U is zero.

From $U(1,0) - U(0,1) = (\alpha - \beta, \alpha' - \beta')$, we deduce that

$$\operatorname{Vol} \mathcal{R}(U(\mathcal{S})) \geq |\alpha - \beta| \, |\alpha' - \beta'| \, .$$

Whenever $|\alpha - \beta| \geq 2$ and $|\alpha' - \beta'| \geq 2$, we are done. Therefore, it remains to examine the following cases:

- If $\alpha = \beta$, then $\alpha\beta' - \beta\alpha' = \alpha\left(\beta' - \alpha'\right) = 1$ implies $\alpha = \beta = 1$ and $\beta' = \alpha' + 1$. A direct calculation yields $\operatorname{Vol} \mathcal{R}(U(\mathcal{S})) = 3(3\alpha' + 2) \geq 4$.
- If $\alpha' = \beta'$, then $\alpha\beta' - \beta\alpha' = \alpha'\left(\alpha - \beta\right) = 1$ implies $\alpha' = \beta' = 1$ and $\alpha = \beta + 1$, and then $\operatorname{Vol} \mathcal{R}(U(\mathcal{S})) = 3(3\beta + 2) \geq 4$, since $\beta \geq 1$ holds in this case.

- If $|\alpha - \beta| = 1$, then we distinguish:
  - if $\beta = \alpha+1$, then the horizontal length of $\mathcal{R}(U(\mathcal{S}))$ is at least $3\alpha+2 \geq 5$,
  - if $\alpha = \beta+1$, then the horizontal length of $\mathcal{R}(U(\mathcal{S}))$ is at least $3\beta+2 \geq 5$.
- If $|\alpha' - \beta'| = 1$, then we distinguish:
  - if $\beta' = \alpha' + 1$ then the height of $\mathcal{R}(U(\mathcal{S}))$ is at least $3\alpha' + 2 \geq 5$,
  - if $\alpha' = \beta' + 1$ then the height of $\mathcal{R}(U(\mathcal{S}))$ is at least $3\beta' + 2 \geq 5$. □

Acknowledgments. We would like to thank the anonymous referees for their helpful comments.

References

[AGL04] F. Abu Salem, S. Gao, and A. G. B. Lauder, Factoring polynomials via polytopes, In *ISSAC '04: Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation*, pages 4–11, New York, 2004, ACM.

[AKS07] M. Avendaño, T. Krick, and M. Sombra, Factoring bivariate sparse (lacunary) polynomials, *J. Complexity*, 23(2):193 – 216, 2007.

[BCS97] P. Bürgisser, M. Clausen, and M. A. Shokrollahi, *Algebraic complexity theory*, volume 315 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*, Springer-Verlag, Berlin, 1997.

[Ber97] L. Bernardin, On square-free factorization of multivariate polynomials over a finite field, *Theoret. Comput. Sci.*, 187(1-2):105–116, 1997.

[Ber98] ———, On bivariate Hensel lifting and its parallelization, In *ISSAC '98: Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation*, pages 96–100, New York, 1998, ACM.

[BHKS09] K. Belabas, M. van Hoeij, J. Klüners, and A. Steel, Factoring polynomials over global fields, *J. Théor. Nombres Bordeaux*, 21(1):15–39, 2009.

[BLS+04] A. Bostan, G. Lecerf, B. Salvy, É. Schost, and B. Wiebelt, Complexity issues in bivariate polynomial factorization, In *ISSAC '04: Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation*, pages 42–49, New York, 2004, ACM.

[Chè10] G. Chèze, A recombination algorithm for the decomposition of multivariate rational functions, Manuscript available from `http://hal.archives-ouvertes.fr/hal-00531601/`, 2010.

[Cox69] H. S. M. Coxeter, *Introduction to Geometry*, John Wiley & Sons Inc., New York, second edition, 1969.

[Gao03] S. Gao, Factoring multivariate polynomials via partial differential equations, *Math. Comp.*, 72:801–822, 2003.

[Gat83] J. von zur Gathen, Factoring sparse multivariate polynomials, In *24th Annual IEEE Symposium on Foundations of Computer Science*, pages 172–179, Los Alamitos, CA, USA, 1983, IEEE Computer Society.

[GG03] J. von zur Gathen and J. Gerhard, *Modern computer algebra*, Cambridge University Press, second edition, 2003.

[GK85] J. von zur Gathen and E. Kaltofen, Factoring sparse multivariate polynomials, *J. Comput. System Sci.*, 31(2):265–287, 1985, Special issue: Twenty-fourth annual symposium on the foundations of computer science (Tucson, Ariz., 1983).

[GR03] S. Gao and V. M. Rodrigues, Irreducibility of polynomials modulo p via Newton polytopes, *J. Number Theory*, 101(1):32–47, 2003.

[GS93] B. Grünbaum and G. C. Shephard, Pick's theorem, *Am. Math. Mon.*, 100(2):150–161, 1993.

[HL10] J. van der Hoeven and G. Lecerf, On the bit-complexity of sparse polynomial and series multiplication, Manuscript available from `http://hal.archives-ouvertes.fr/hal-00476223/`, 2010.

[Kal85] E. Kaltofen, Sparse Hensel lifting, In *EUROCAL '85, Vol. 2 (Linz, 1985)*, volume 204 of *LNCS*, pages 4–17, Springer-Verlag, 1985.

[Kal89] ———, Factorization of polynomials given by straight-line programs, In S. Micali, editor, *Randomness and Computation*, volume 5 of *Advances in Computing Research*, pages 375–412, JAI, 1989.

[KK06] E. Kaltofen and P. Koiran, Finding small degree factors of multivariate supersparse (lacunary) polynomials over algebraic number fields, In *ISSAC '06: Proceedings of the 2006 International Symposium on Symbolic and Algebraic Computation*, pages 162–168, New York, 2006, ACM.

[Lec06] G. Lecerf, Sharp precision in Hensel lifting for bivariate polynomial factorization, *Math. Comp.*, 75:921–933, 2006.

[Lec07] ———, Improved dense multivariate polynomial factorization algorithms, *J. Symbolic Comput.*, 42(4):477–494, 2007.

[Lec08] ———, Fast separable factorization and applications, *Appl. Algebr. Engrg. Comm. Comput.*, 19(2), 2008.

[Lec10] ———, New recombination algorithms for bivariate polynomial factorization based on Hensel lifting, *Appl. Algebr. Engrg. Comm. Comput.*, 21(2):151–176, 2010.

[Ost21] A. M. Ostrowski, Über die Bedeutung der Theorie der konvexen Polyeder für die formale Algebra, *Jahresber. Deutsch. Math.-Verein.*, 30(2):98–99, 1921, Talk given at *Der Deutsche Mathematikertag vom 18–24 September 1921 in Jena.*

[Ost75] ———, On multiplication and factorization of polynomials, I. Lexicographic orderings and extreme aggregates of terms, *Aequationes Math.*, 13(3):201–228, 1975.

[Ost99] ———, On the significance of the theory of convex polyhedra for formal algebra, *SIGSAM Bull.*, 33(1):5, 1999, Translated from [Ost21].

[Pot08] A. Poteaux, *Calcul de développements de Puiseux et application au calcul du groupe de monodromie d'une courbe algébrique plane*, PhD thesis, Université de Limoges, 2008, Manuscript available from `http://www.ag.jku.at/~adrien`.

[PR10] A. Poteaux and M. Rybowicz, Complexity bounds for the rational Newton-Puiseux algorithm over finite fields, *Appl. Algebra Engrg. Comm. Comput.*, (to appear) 2010.

[PS85] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag, 1985.

[Ser96] J.-P. Serre, *A Course in Arithmetic*, Springer-Verlag, New York, 1996, Translated from the French, Graduate Texts in Mathematics, No. 7.

[Ste05] A. Steel, Conquering inseparability: primary decomposition and multivariate factorization over algebraic function fields of positive characteristic, *J. Symbolic Comput.*, 40(3):1053–1075, 2005.

[Wei10] M. Weimann, A lifting and recombination algorithm for rational factorization of sparse polynomials, *J. Complexity*, 26(6):608–628, 2010.

[Zip79] R. Zippel, Probabilistic algorithms for sparse polynomials, In *EUROSAM '79: Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 216–226, Springer-Verlag, 1979.

[Zip81] R. Zippel, Newton's iteration and the sparse Hensel algorithm (Extended Abstract), In *SYMSAC '81: Proceedings of the fourth ACM Symposium on Symbolic and Algebraic Computation*, pages 68–72, New York, 1981, ACM.

[Zip93] ———, *Effective Polynomial Computation*, Kluwer Academic Publishers, 1993.

Laboratoire d'Informatique UMR 7161 CNRS, École polytechnique, Route de Saclay, 91128 Palaiseau Cedex, France
E-mail address: `berthomieu@lix.polytechnique.fr`
URL: `http://www.lix.polytechnique.fr/~berthomieu`

Laboratoire d'Informatique UMR 7161 CNRS, École polytechnique, Route de Saclay, 91128 Palaiseau Cedex, France
E-mail address: `gregoire.lecerf@math.cnrs.fr`
URL: `http://lecerf.perso.math.cnrs.fr`