FOUNDATIONS ADVANCES

ISSN 2053-2733

A fast algorithm to find reduced hyperplane unit cells and solve *N*-dimensional Bézout's identities

Cyril Cayron*

Laboratory of Thermo Mechanical Metallurgy (LMTM), PX Group Chair, EPFL, Rue de la Maladière 71b, Neuchâtel, 2000, Switzerland. *Correspondence e-mail: cyril.cayron@epfl.ch

Received 1 February 2021
Accepted 1 July 2021

Edited by L. Palatinus, Czech Academy of Sciences, Czech Republic

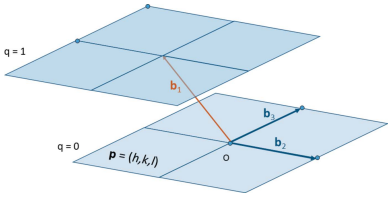
Keywords: *N*-dimensional Bézout's identity; hyperplane unit cell; integer relation; twinning.

Deformation twinning on a plane is a simple shear that transforms a unit cell attached to the plane into another unit cell equivalent by mirror symmetry or 180° rotation. Thus, crystallographic models of twinning require the determination of the short unit cells attached to the planes, or hyperplanes for dimensions higher than 3. Here, a method is presented to find them. Equivalently, it gives the solutions of the *N*-dimensional Bézout's identity associated with the Miller indices of the hyperplane.

1. Introduction

How to determine a unit cell attached to a plane $\mathbf{p} = (h, k, l)$? This problem occurs for example in the crystallographic models of twinning, when the obliquity or the shear values must be calculated for many planes. It is intuitively solved for low-index planes, but the solutions are more difficult to obtain for high-index planes. In addition, if a unit cell can be found, can it be reduced to a smaller one? In dimension N, the difficulty of finding a small unit cell attached to a hyperplane of dimension $N-1$ becomes even more pronounced. Let us express mathematically this 'hyperplane unit-cell problem' by the notations detailed in Appendix *A*. We assume that a hyperplane is known only by its Miller indices h, k, l which are coprime integers, or equivalently by its normal which is expressed as an integer vector of coordinates h, k, l in the reciprocal space. We want to determine a small unit cell such that one short integer vector of the cell points to a node of the first layer parallel to the hyperplane, and the other $N-1$ short integer vectors lie in the hyperplane. The 'out-of-plane' vector and the 'in-plane' vectors are noted $\mathbf{b}_1$ and $\mathbf{b}_2, \ldots, \mathbf{b}_j, \ldots, \mathbf{b}_N$, respectively. The vector $\mathbf{b}_1$ is such that $\mathbf{p}^t\mathbf{b}_1 = 1$, and the $N-1$ vectors $\mathbf{b}_2, \ldots, \mathbf{b}_j, \ldots, \mathbf{b}_N$ are such that $\mathbf{p}^t\mathbf{b}_j = 0$. The coordinates of the vector $\mathbf{b}_1$ constitute a solution of the *N*-dimensional Bézout's identity formed on the coordinates of $\mathbf{p}$. The coordinates of any of the vectors $\mathbf{b}_j$, $j \in \{2, \ldots, N\}$, are solutions of what is called an 'integer relation' with the coordinates of $\mathbf{p}$ (Appendix *A*). For example, with $N = 3$, the integer coordinates u, v, w of $\mathbf{b}_1$ verify the equation $uh + vk + wl = 1$; the integer coordinates u, v, w of the vector $\mathbf{b}_2$ (or $\mathbf{b}_3$) verify the equation $uh + vk + wl = 0$, as illustrated in Fig. 1.

Finding solutions to integer relations is not complicated. For $N = 3$, if we know a plane $\mathbf{p} = (h, k, l)$ with let us say $k \neq 0$, it is not difficult to find two integer vectors $\mathbf{b}_2$ and $\mathbf{b}_3$ in this plane, for example, $\mathbf{b}_2 = [-k, h, 0]$ and $\mathbf{b}_3 = [0, -l, k]$. The difficult part of the problem is to find vectors with small coordinates by considering all the possible linear combinations of the Miller indices h, k, l. Finding the shortest solutions in dimension N is an NP-hard (non-deterministic polynomial-



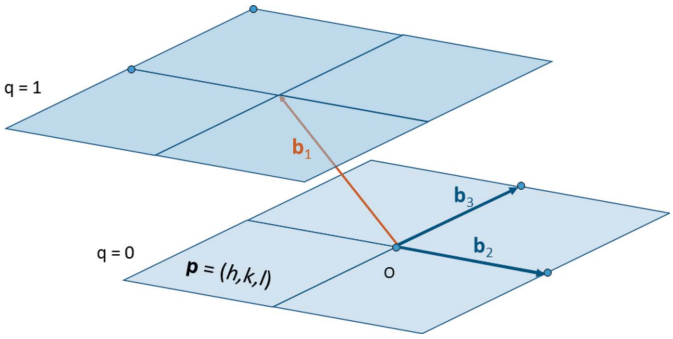


Figure 1
Unit cell associated with the plane $\mathbf{p} = (h, k, l)$. The out-of-plane vector $\mathbf{b}_1$ points to a node of the layer $q = 1$, and the in-plane vectors $\mathbf{b}_2$ and $\mathbf{b}_3$ lie in the layer $q = 0$. The vector $\mathbf{b}_1$ is a solution of the Bézout's identity $\mathbf{p}^t\mathbf{b}_1 = 1$, and the vectors $\mathbf{b}_2$ and $\mathbf{b}_3$ are solutions of the integer relations $\mathbf{p}^t\mathbf{b}_2 = 0$ and $\mathbf{p}^t\mathbf{b}_3 = 0$.

time) problem well known in computer science and cryptography. An algorithm called PSLQ gives short solutions to integer relations with any vector $\mathbf{p} \in \mathbb{R}^N$ (Ferguson *et al.*, 1999; see also Wikipedia, 2021*a*). It has permitted the discovery of numerous previously unknown identities among real numbers; one of them is the formula that allows the calculation of the nth hexadecimal digit of π without computing the preceding digits (Bailey *et al.*, 1997; Raayoni *et al.*, 2021). The algorithm presented in the present paper gives only solutions for the vectors $\mathbf{p} \in \mathbb{Z}^N$ but, as we will show, the vectors we obtain are shorter than those obtained by PSLQ. Our algorithm actually provides simultaneously a short solution to the N-dimensional Bézout's identity and short solutions to the 'integer relations'. It gives the affine space of all the solutions of the N-dimensional Bézout's identity. From a crystallographic point of view, it provides a small unit cell attached to a hyperplane.

Recently, Gorfman (2020) proposed a method to find some solutions to an intermediate problem that we will call the 'column-constrained unimodular matrix' (CCUM) problem in order to differentiate it from the initial 'hyperplane unit-cell problem'. The CCUM problem consists of finding a unimodular matrix $\mathbf{M}$ such that the first column is equal to a fixed integer vector $\mathbf{t}$. We recall that a unimodular matrix has integer entries and its determinant is ± 1. Note that, in Gorfman's paper, it was the last vector (and not the first one) that was imposed, but that does not change the problem. Gorfman's approach involves a series of multiplication with matrices called $\mathbf{S}$ containing 0, 1 and -1 in order to reduce the imposed vector $\mathbf{t}$ to a unit vector (a vector for which one of its coordinates is 1 and the others are 0). Gorfman showed that the same algorithm applied in the reciprocal space to a vector $\mathbf{p}$ gives a solution to the hyperplane unit-cell problem. Let us explain how it works with our notations. For an imposed reciprocal vector $\mathbf{p}$, Gorfman's method permits one to obtain a unimodular matrix $\mathbf{M}^*$ that has $\mathbf{p}$ for the first column vector. Then, the inverse of its transpose $\mathbf{M} = (\mathbf{M}^*)^{-t}$ is calculated. Since $\mathbf{M}^*$ is a unimodular matrix, the matrix $\mathbf{M}$ is also unimodular, which implies that its columns are integer vectors. Let us call them $\mathbf{b}_j$. Since $(\mathbf{M}^*)^t\mathbf{M}$ is the identity matrix, its first column is a vector that has 1 as the first coordinate and 0 for all the other coordinates. This means that $\mathbf{p}^t\mathbf{b}_1 = 1$ and $\mathbf{p}^t\mathbf{b}_j = 0$ for $j \in \{2, \ldots, N\}$, which proves that the matrix $\mathbf{M}$ is a solution of the hyperplane unit-cell problem. Gorfman's idea of using unimodular matrices is very interesting and his approach is innovative and inspiring, but it does not give short solutions. For example, for the plane $\mathbf{p}^t = (12, 20, 225)$, the solution determined by his algorithm in which the first imposed column vector is $\mathbf{p}$ is

$$\mathbf{M}^* = \begin{bmatrix} 12 & 2 & 3 \\ 20 & 3 & 5 \\ 225 & 4 & 56 \end{bmatrix}.$$

The inverse of its transpose is

$$\mathbf{M} = \begin{bmatrix} 148 & 5 & -595 \\ -100 & -3 & 402 \\ 1 & 0 & -4 \end{bmatrix}.$$

The reader can check that the scalar product of $(12, 20, 225)$ with the first column vector $[148, -100, 1]$ is 1, and that the scalar product with the last column vectors $[-595, 402, -4]$ and $[5, -3, 0]$ is 0. However, the vector $[148, -100, 1]$ solution of the 3D Bézout identity and the vector $[-595, 402, -4]$ solution of the integer relation are large. The vector $[-595, 402, -4]$ is even larger than the obvious solution $[0, -4, 45]$. More generally, Gorfman suggests that the algorithm could be 'an alternative approach to calculate the Bezout coefficients', but we would like to show that the opposite approach is possible. The aim of the paper is to show that determination of the Bézout's coefficients is an efficient way to find short solutions of both the CCUM problem and hyperplane unit-cell problem. The algorithm proposed in the present paper is based on Euclidean division. An algorithm to determine some short solutions to the N-dimensional Bézout's identity is proposed in Section 2. The algorithm to solve the CCUM problem is detailed in Section 3. Sections 2 and 3 are independent. In Section 4, we explain how to combine the two algorithms to find short solutions to the hyperplane unit-cell problem. Some examples will be given and compared with the PSLQ algorithm. The method has been encoded in a Python 3.8 computer program called *GeneralizedBezout*. The examples given were obtained on a laptop computer equipped with an Intel Core i7-4600 CPU 2.1 GHz, 64-bit Windows system with a RAM of 8 GB. Note: the Python program *GeneralizedBezout* is freely available from the author upon request.

2. N-dimensional Bézout's identity

Given a set of integers $\{p_i, i = 1, \ldots, N\}$ we look for another set of integers $\{u_i, i = 1, \ldots, N\}$ such that $\sum_{i=1}^{N} p_i u_i = 1$. In other words, given an integer vector $\mathbf{p}$ of coordinates p_i, we want to get the coordinates u_i of an integer vector $\mathbf{u}$ that is such that $\mathbf{p}^t\mathbf{u} = 1$. If $N = 2$, the fast and well known algorithm based on Euclidean division gives a solution that is also the shortest one (Capparelli, 2020; Wikipedia, 2021*b*). Surprisingly, we could not find in the literature algorithms in high dimensions N. We propose here two recursive algorithms.

They give different solutions that are all valuable, but we will see that the second one gives shorter solutions.

Method-0. We consider p_1 and p_2 the two first coordinates of **p**, and we call (u, v) their Bézout numbers, *i.e.* $up_1 + vp_2 = \gcd(p_1, p_2)$. If we note $\{k_i, i = 2, \ldots, N\}$ the Bézout numbers in dimension $N-1$ associated with the set $\{\gcd(p_1, p_2), p_3, \ldots, p_N\}$, a solution of the N-dimensional Bézout's identity is thus $\{uk_2, vk_2, k_3, \ldots, k_N\}$. This method is easy to compute by recursion until the dimension decreases to $N = 2$ for which the solution is given by the classical Bézout's algorithm. The problem related to this method is that the absolute values of the Bézout numbers u_i can be quite high. One could screen all the pairs (p_i, p_j) in place of (p_1, p_2) to determine the lowest Bézout numbers but this method would be unrealistic for high dimensions N. We could find another method for which the values are lower than those determined by method-0.

Method-1. We consider the set of integers $\{p_i, i = 1, 2, \ldots, N\}$. If $\exists i$, $|p_i| = 1$, the solution of the Bézout identity is immediately $\{0, \ldots 0, p_i, 0 \ldots, 0\}$. If none of the p_i's has 1 as absolute value, the set $\{p_i\}$ is sorted in decreasing order of the absolute values of p_i. The sorting permutation σ is kept in memory. The smaller non-null value is called p_{i_0}. We calculate the quotient set and the residue set $\{q_i, i = 1, 2, \ldots, i_0 - 1\}$ and $\{r_i, i = 1, 2, \ldots, i_0 - 1\}$ with $q_i = \lfloor p_i/p_{i_0} \rfloor$ and $r_i = p_i - q_i p_{i_0}$, quotient and remainder of the Euclidean division by p_{i_0}. If we note $\{u_1, u_2, \ldots, u_{i_0-1}, 0, \ldots, 0\}$ the Bézout numbers associated with the set $\{r_1, r_2, \ldots, r_{i_0-1}, 0, \ldots, 0\}$, a solution of the N-dimensional Bézout's identity is $\{u_1, u_2, \ldots, u_{i_0-1}, -\sum_{i=1}^{i_0-1} q_i u_i, 0, \ldots, 0\}$. This method is easy to compute by recursion until one of the absolute values of the input vector is 1. The correct order of the Bézout numbers associated with the initial set $\{p_i, i = 1, 2, \ldots, N\}$ is restored by applying σ^{-1}. The pseudo-code is given in Fig. 2.

The Bézout numbers calculated with method-1 are smaller than those obtained by method-0. Only method-1 will be considered in the rest of the paper. With the vector $\mathbf{p}^{\mathrm{t}} = (12, 20, 225)$, it gives $\mathbf{u} = [-17, -1, 1]$. With the vector $\mathbf{p}^{\mathrm{t}} = (51, 450, -102, 240, -277, 54, 450, 532)$, it gives $\mathbf{u} = [-3, 0, 0, 0, 0, -3, 0, 1]$. The calculation lasts only a few ms. Even if method-1 gives small Bézout vectors **u**, it may not give systematically the smallest ones. We will see in Section 4 how 'hyperplane shearing' can give shorter Bézout vectors **u** with the help of the CCUM algorithm detailed in Section 3.

Function *Bezout* **(p)**

Input: **p** is a N-dimensional vector of coprime coordinates $\mathbf{p} = (p_1, p_2, \ldots, p_N)$
Output: $\mathbf{b}_1$ solution of Bezout's identity $\mathbf{p}^{\mathrm{t}}\mathbf{b}_1 = 1$

If $\exists i, |p_i| = 1$, **return** $\{0, \ldots 0, p_i, 0 \ldots, 0\}$

Else

Sort $(p_1, p_2, \ldots, p_N)$ in decreasing order of $|p_i|$. σ = sorting permutation.
Determine the first non-null index i_0 such that $p_i = 0$ *for* $i > i_0$
Calculate $q_i = \left\lfloor \frac{p_i}{p_{i_0}} \right\rfloor$ and $r_i = p_i - q_i p_{i_0}$
$\mathbf{r} = \{r_1, r_2, \ldots, r_{i_0-1}, 0, \ldots, 0\}$
Calculate $\{u_1, u_2, \ldots, u_{i_0-1}, 0, \ldots, 0\} = Bezout\ (\mathbf{r})$
return $\sigma^{-1}\left\{u_1, u_2, \ldots, u_{i_0-1}, -\sum_{i=1}^{i_0-1} q_i u_i\right\}\sigma$

Figure 2
Pseudocode to find Bézout numbers associated with the coordinates of a vector **p**.

3. Algorithm to solve the column-constrained unimodular matrix problem

3.1. Case where one of the coordinates of t is ±1

Now we consider the CCUM problem. There is a simple and immediate solution if the first coordinate of **t** is 1. In that case, any diagonal or even triangular matrix **M** with 1 in the diagonal and with **t** as the first column checks the condition $\det(\mathbf{M}) = 1$. If the first coordinate of **t** is −1, changing one 1 into −1 in the diagonal is sufficient to maintain $\det(\mathbf{M}) = 1$. The example used by Gorfman (2020) with the vector **t** of coordinates [−1, 4, 2] enters in this category. A direct solution is

$$\mathbf{M} = \begin{bmatrix} -1 & 0 & 0 \\ 4 & 1 & 0 \\ 2 & 0 & -1 \end{bmatrix}.$$

Note that the result is obtained without any calculation. If one of the coordinates of **t** is 1 in a position $i > 1$, then a simple matrix of permutation **P** is sufficient to recalculate the matrix **M**. We will not give more details here because the solutions are actually included in the more general method based on Bézout's identity explained as follows.

3.2. Case where t has at least one pair of coprime coordinates

In the case $N = 2$, the general solution to the CCUM problem is given by the classical 2D Bézout's identity. We note

$$\mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

the imposed vector. There is a solution if and only if the integers t_1 and t_2 are coprime, and the solution is simply

$$\begin{bmatrix} t_1 & -v \\ t_2 & u \end{bmatrix},$$

where u, v are the Bézout numbers associated with t_1 and t_2, *i.e.* solutions of the equation $ut_1 + vt_2 = 1$. If t_1 and t_2 are not coprime, the determinant of any matrix **M** with **t** in the first column would be a multiple of $\gcd(t_1, t_2)$, the greatest common divisor of t_1 and t_2, and thus cannot be equal to ±1. The resulting vector

$$\begin{bmatrix} -v \\ u \end{bmatrix}$$

is the shortest vector. The other vectors are

$$\begin{bmatrix} -v + kt_1 \\ u + kt_2 \end{bmatrix}$$

with $k \in \mathbb{Z}$.

Now, we consider the case where $N > 2$ and the vector **t** has its two first coordinates t_1 and t_2 that are coprime numbers. We consider the matrix **M** made of two blocks, the top left one is

$$\begin{bmatrix} t_1 & -v \\ t_2 & u \end{bmatrix},$$

where u, v are the Bézout numbers associated with t_1 and t_2, and the bottom right one is the $(N-2)\times(N-2)$ identity matrix. Then, the first column of **M** is replaced by **t** (t_1 and t_2 are not changed, and the zeros in $\mathbf{M}_{i,1}$ are replaced by t_i, $i > 2$). The matrix **M** is the solution of the CCUM problem.

When the two coprime coordinates of vector **t**, t_1 and t_2, are not the first ones, the permutation matrices $\mathbf{P}(i, 1)$ and $\mathbf{P}(j, 2)$ are used to return to the previous case. We recall that a permutation matrix $\mathbf{P}(i, j)$ is a $N \times N$ identity matrix, except for the line i for which 1 is written in the column j, and for the column j where 1 is written in the line i. Permutation matrices are unimodular matrices and are equal to their inverse. The unimodular matrix $\mathbf{P} = \mathbf{P}(i, 1)\mathbf{P}(j, 2)$ is such that the vector $\mathbf{P}\cdot\mathbf{t}$ has for first coordinates the coprime numbers t_1 and t_2. We thus return to the previous case. If we call **M** the two-block solution of that case, the solution of the problem is given by the matrix $\mathbf{P}^{-1}\mathbf{M}$. Note that $\mathbf{P}^{-1} = \mathbf{P}(j, 2)\mathbf{P}(i, 1) \neq \mathbf{P}$.

With **t** of coordinates [12, 20, 225], the algorithm gives

$$\mathbf{M} = \begin{bmatrix} 12 & 1 & 0 \\ 20 & 2 & 1 \\ 225 & 0 & -56 \end{bmatrix}.$$

The algorithm works very efficiently, even in high dimensions and with large coordinates. For example, with **t** of coordinates [1551, −540, 67, −102, 2140, −277, 32, 366, 450, 1532], the algorithm gives immediately (less than 1 ms) a solution:

$$\mathbf{M} = \begin{bmatrix} 1551 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -463 \\ -540 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 67 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -20 \\ -102 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2140 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -277 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 32 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 366 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 450 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1532 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Function *CCUM* (**t**)

Input: **t** is a N-dimensional vector of coprime coordinates $(t_1, t_2, \ldots, t_N)$
Output: **M** integral matrix such that $det(\mathbf{M})=1$ and first column $\mathbf{M}_{i,1} = \mathbf{t}$

If $\exists i, t_i = \pm 1$,
$\mathbf{M} = N \times N$ *identity matrix*
$\mathbf{M}_{i,1} \leftarrow \mathbf{t}$
If $t_i = -1, \mathrm{M}_{N,N} \leftarrow$ -1
return M

Elif (t_1, t_2) coprime,
(u, v) = Bezout($[t_1, t_2]$)

$$\textbf{return M} = \begin{pmatrix} t_1 & -v & 0 & 0 & \ldots & 0 \\ t_2 & u & 0 & 0 & \ldots & 0 \\ t_3 & 0 & 1 & 0 & \ldots & 0 \\ t_4 & 0 & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & 0 & \ddots & 0 \\ t_N & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Elif $\exists (i, j)$, (t_i, t_j) coprime
$\sigma = \mathbf{P}(j, 2)\ \mathbf{P}(i, 1)$
M = *CCUM* (σ **t**) (*treated by the 1st Elif*)
return $\sigma^{-1}\mathbf{M}$

Else
(u, v) = Bezout($[t_1, t_2]$)
(α, β) = Bezout($[u, v]$)

$$\mathbf{K} = \begin{pmatrix} u & v & 0 & 0 & \ldots & 0 \\ -\beta & \alpha & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

L = *CCUM* (**K t**) (*treated by the 2st Elif*)
return $\mathbf{K}^{-1}\mathbf{L}$

Figure 3
Pseudocode to find a column-constrained unimodular matrix associated with an integer vector **t**.

3.3. Case where none of the pairs of coordinates of t are coprime

Now, let us consider the rarer cases in which none of the pairs (t_i, t_j) of coordinates of **t** are coprime despite the fact that the set of coordinates of **t** are coprime (as mentioned previously, if they are not, there is no solution to the problem). The set of integers $\{t_i, i = 1, \ldots, N\}$ is said to be 'coprime but not pairwise coprime'. A classical example is {6, 10, 15}. Let us recall that in large dimensions N, the probability that a set of integers $\{t_i\}$ is coprime but not pairwise coprime is very small because the probability that two randomly chosen integers are coprime is quite high: it is equal to $1/[\zeta(2)] = 6/\pi^2 \simeq 61\%$, where ζ refers to the Riemann zeta function (Wikipedia, 2021*c*). The exact calculation of the probability for a set of N integers $\{t_i\}$ to be coprime but not pairwise coprime as a function of N is not straightforward and is beyond the scope of the present study. Even if rare, these cases can be solved as follows. We consider the two first coordinates t_1 and t_2 of the vector **t** (any pair of coordinates would also work). As t_1 and t_2 are not coprime, they can be written $t_1 = xy$ and $t_2 = yz$, where x, y, z are three integers and $y = \gcd(t_1, t_2) > 1$. It is important to note here that there is at least another coordinate t_i with $i > 2$ that cannot be divided by y because if it were not so the set $\{t_i\}$ would not be coprime. We call (u, v) the Bézout numbers associated with (t_1, t_2), $ut_1 + vt_2 = y$. The pair (u, v) are also the Bézout numbers associated with (x, z), $ux + vz = 1$, *i.e.* (u, v) are also coprime. We call (α, β) the Bézout numbers associated with (u, v). We consider the matrix

$$\begin{bmatrix} u & v \\ -\beta & \alpha \end{bmatrix},$$

its determinant is 1, and

$$\mathbf{B}\cdot\begin{bmatrix} xy \\ yz \end{bmatrix} = \begin{bmatrix} y \\ ky \end{bmatrix},$$

with $k \in \mathbb{Z}$. We build the $N \times N$ matrix $\mathbf{K}$ from the 2×2 block $\mathbf{B}$ and from the $(N-2) \times (N-2)$ identity matrix. The first coordinate $(\mathbf{K}\cdot\mathbf{t})_{(1)}$ of the new vector $\mathbf{K}\cdot\mathbf{t}$ is coprime with at least one of the coordinates $(\mathbf{K}\cdot\mathbf{t})_{(i)}$ with $i > 2$. It means that the method described in Section 3.2 can be applied to calculate a matrix $\mathbf{L}$ such that $\det(\mathbf{L}) = 1$, and its last column is the vector $\mathbf{K}\cdot\mathbf{t}$. The matrix $\mathbf{M} = \mathbf{K}^{-1}\mathbf{L}$ is then such that its determinant is also 1 and its last column is $\mathbf{t}$. As the determinant of $\mathbf{K}$ is 1, $\mathbf{K}^{-1}$ is the adjugate (transpose of the cofactor matrix) of $\mathbf{K}$, and is thus an integer matrix. Consequently, $\mathbf{M}$ is also an integer matrix, solution of the problem.

The algorithm is effective and fast, whatever the dimension N of the vector $\mathbf{t}$. The pseudocode is shown in Fig. 3.

We give an example with the classical set of coprime but not coprime coordinates [6, 10, 15]. The algorithm gives immediately a solution (the vectors are written in columns):

$$\mathbf{M} = \begin{bmatrix} 6 & 1 & 0 \\ 10 & 2 & 1 \\ 15 & 0 & -7 \end{bmatrix}.$$

Let us build another example with a vector $\mathbf{t}$ of coordinates $[-42, 10, 15, -30, 6]$. A solution is

$$\mathbf{M} = \begin{bmatrix} -42 & 0 & 4 & 0 & 1 \\ 10 & 0 & -1 & 0 & 0 \\ 15 & 0 & 0 & 0 & -7 \\ -30 & -1 & 0 & 0 & 0 \\ 6 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

4. Hyperplane unit cell by oblique projection

Let us recall the hyperplane unit-cell problem. We are looking for a set of N vectors $\{\mathbf{b}_1, \ldots, \mathbf{b}_j, \ldots, \mathbf{b}_N\}$ such that the first out-of-plane vector $\mathbf{b}_1$ is such that $\mathbf{p}^{\mathrm{t}}\mathbf{b}_1 = 1$ (pointing to a node of the layer $q = 1$), and the $N-1$ in-plane vectors $\mathbf{b}_2, \ldots, \mathbf{b}_j, \ldots, \mathbf{b}_N$ are such that $\mathbf{p}^{\mathrm{t}}\mathbf{b}_j = 0$ (lying in the layer $q = 0$). The solution $\mathbf{b}_1$ of Bezout's identity is placed in the first position to be coherent with the notations we used in a separate paper dedicated to the lattice reduction (Cayron, 2021). The method we propose uses a short solution to the N-dimensional Bézout identity (Section 2) and a solution of the CCUM problem (Section 3).

We start from the input vector $\mathbf{p}$. The coordinates of the vector $\mathbf{b}_1$ pointing to a node of the layer $q = 1$ are the solution of the Bézout's identity associated with $\mathbf{p}$. They are obtained by the algorithm detailed in Section 2 (method-1). Now, how to determine the $N-1$ vectors in the layer $q = 0$? We consider the unimodular matrix $\mathbf{M}$ that is such that the first column is the vector $\mathbf{b}_1$. The $N-1$ next column vectors of the matrix $\mathbf{M}$ are called $\mathbf{v}_j$ for $j \in \{2, \ldots N\}$. Each of these vectors belongs to the lattice; thus, they verify $\mathbf{p}^{\mathrm{t}}\mathbf{v}_j = q_j \in \mathbb{Z}$. The vectors $\mathbf{b}_j = \mathbf{v}_j - q_j\mathbf{b}_1$ verify $\mathbf{p}^{\mathrm{t}}\mathbf{b}_j = 0$ for $j \in \{2, \ldots N\}$; *i.e.* they are in-plane vectors lying in the layer $q = 0$. Geometrically, the vectors $\mathbf{b}_j$ are obtained by oblique projection of the vectors $\mathbf{v}_j$ along $\mathbf{b}_1$ onto the plane $\mathbf{p}$, as illustrated for $N = 3$ in Fig. 4.

Now, we have a cell $\mathbf{U} = (\mathbf{b}_1, \ldots, \mathbf{b}_j, \ldots, \mathbf{b}_N)$ attached to the plane $\mathbf{p}$ such that $\det(\mathbf{U}) = 1$, $\mathbf{p}^{\mathrm{t}}\mathbf{b}_1 = 1$ and $\mathbf{p}^{\mathrm{t}}\mathbf{b}_j = 0$ for $i \in \{2, \ldots N\}$. It is thus the unit cell we were looking for. As the vectors used for the projection are short, the unit cell is not large. It can be reduced even more. There are different methods to find a reduced unit cell $\mathbf{U}' = (\mathbf{b}'_1, \ldots, \mathbf{b}'_j, \ldots, \mathbf{b}'_N)$, with $\mathbf{b}'_j$ that have the same properties as $\mathbf{b}_j$ with the vector $\mathbf{p}$, but with shorter lengths and with angles between them closer to orthogonality. One could apply for example the LLL algorithm well known in computer science (Lenstra *et al.*, 1982). We realized however that the algorithm developed in Section 4 can also be used to define the operation of 'hyperplane shearing' which consists of shearing the unit cell such that the vector $\mathbf{b}_1$ becomes $\mathbf{b}'_1$ nearly normal to the plane $\mathbf{p}$ as illustrated in Fig. 4, and that this operation can be coupled with other lattice reduction methods to rival LLL implemented in *Mathematica*. The hyperplane reduction and its application to lattice reduction are detailed in a separate paper (Cayron, 2021). The pseudocode of the set of operations Bézout–CCUM–Projection–Hyperplane reduction is shown in Fig. 5.

The program written in Python 3.8 called *GeneralizedBezout* incorporates the lattice reduction operation described by Cayron (2021). Let us give some examples we obtained:

(i) With $\mathbf{p}^{\mathrm{t}} = (12, 20, 225)$. The Bézout vector associated with the plane $\mathbf{p}$ given by method-1 described in Section 2 is $\mathbf{b}_1 = [-17, -1, 1]$. After determining a first unit cell by projections along $\mathbf{b}_1$, and after lattice reduction, this vector becomes $\mathbf{b}'_1 = [-7, -7, 1]$. The final reduced unit cell is given by the matrix

$$\mathbf{U}' = \begin{bmatrix} -7 & -5 & 20 \\ -7 & 3 & 33 \\ 1 & 0 & -4 \end{bmatrix},$$

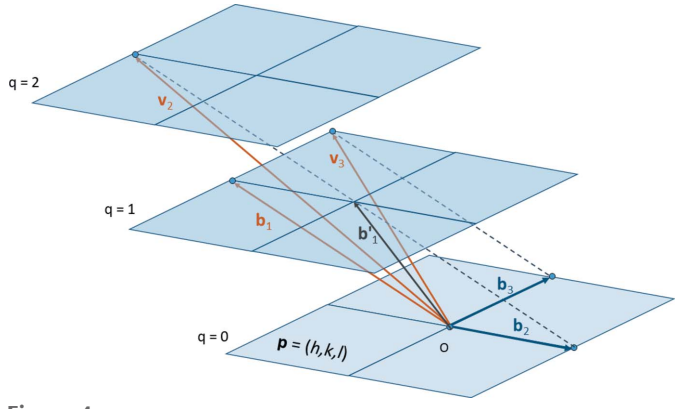


Figure 4
Projections of the vectors $\mathbf{v}_2$, $\mathbf{v}_3$ along $\mathbf{b}_1$ onto the plane $\mathbf{p} = (h, k, l)$. The vector $\mathbf{b}_1$ is a short solution of the Bézout's identity $\mathbf{p}_1^{\mathrm{t}}\mathbf{b}_1 = 1$. The vectors $\mathbf{v}_2$, $\mathbf{v}_3$ are the solutions of the $\mathbf{b}_1$–CCUM problem. The vector $\mathbf{b}_1$ is a short vector that can be further shortened into a vector $\mathbf{b}'_1$ by 'hyperplane shearing' (Cayron, 2021).

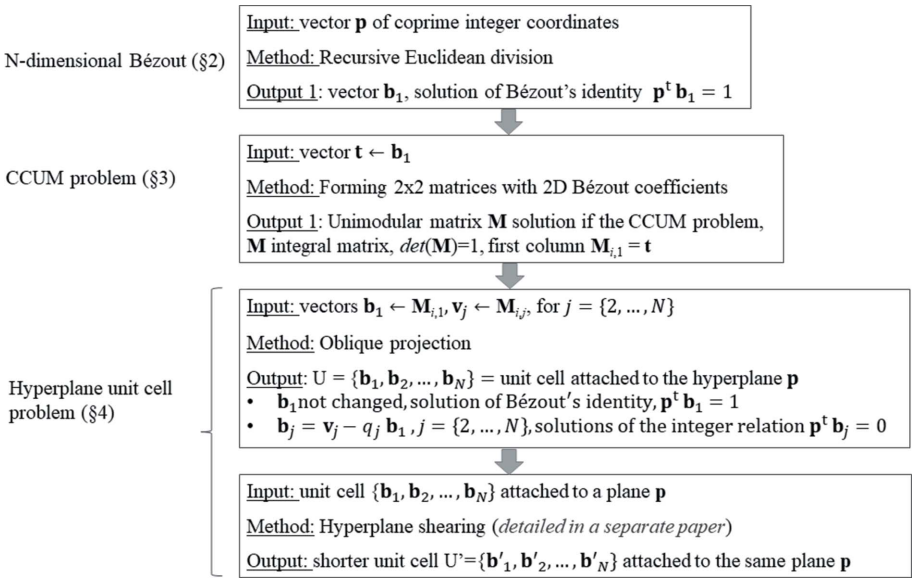


Figure 5
Pseudocode of the sequence of operations to determine a short unit cell associated with a hyperplane **p**. The unit cell is made of N short vectors $\mathbf{U}' = (\mathbf{b}'_1, \ldots, \mathbf{b}'_j, \ldots, \mathbf{b}'_N)$, such that $\mathbf{p}^t\mathbf{b}'_1 = 1$ and $\mathbf{p}^t\mathbf{b}'_j = 0$.

where the vectors are written in columns. The first vector is the short solution of the 3D Bézout's identity and the other vectors are short solutions of the integer relations with the coordinates of **p**.

(ii) With $\mathbf{p}^t$ = (−54, 131, −48, 632, 23, 177, 333, 99, −581, 377). The coordinates were randomly chosen. The Bézout vector associated with the plane **p** given by method-1 described in Section 2 is $\mathbf{b}_1 = [1, 0, 0, 0, 11, 0, 0, -2, 0, 0]$. After determining a first unit cell by projections parallel to $\mathbf{b}_1$, and after reducing this unit cell, this vector becomes $\mathbf{b}'_1 = [0, 1, 1, 0, 1, 0, 0, 1, 1, 1]$. The reduced unit cell is

$$\mathbf{U}' = \begin{bmatrix} 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & -2 & 1 & -1 & 0 & 0 & -1 \\ 1 & -2 & -2 & -1 & -2 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & -1 & 1 & 2 & 0 & 0 & -1 & 1 \\ 0 & -2 & 1 & 0 & 0 & -1 & 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & 1 & -1 & 0 & 0 & -1 & -1 & 1 \\ 1 & 0 & -2 & 0 & 1 & 0 & -2 & 0 & -1 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & -1 & -2 & 1 & 0 & 1 & -1 & 0 & 0 \end{bmatrix},$$

where the vectors are written in columns. The first vector is a short solution of the 10D Bézout's identity and the other vectors are short solutions to the integer relations with the coordinates of **p**. The calculation lasted 20 ms. The PSLQ method implemented in *Mathematica* under the function *FindIntegerNullVector* gives only one solution which is $[1, 0, -2, 0, 2, 0, 2, 0, 0, -2]$. We notice that this vector is larger than all the vectors $\mathbf{b}'_j$ in columns $j \in \{2, \ldots N\}$ of the matrix $\mathbf{U}'$.

The matrix $\mathbf{U}'$ is interpreted crystallographically/geometrically as the unit cell attached to the hyperplane **p**. From an algebraic point of view, $\mathbf{U}' = (\mathbf{b}'_1, \ldots, \mathbf{b}'_i, \ldots, \mathbf{b}'_N)$ can equivalently be understood as the infinite set of solutions of the N-dimensional Bézout's identity, where $\mathbf{b}'_1$ is a short solution of the equation $\mathbf{p}^t\mathbf{b}'_1 = 1$, and the other vectors are short solutions of the integer relation $\mathbf{p}^t\mathbf{b}'_j = 0$, $j \in \{2, \ldots N\}$. The set of solutions of Bézout's identity is thus $\mathbf{b}'_1 + \{\mathbb{Z}.\ \mathbf{b}'_j\}$ with $j \in \{2, \ldots N\}$, where $\{\mathbb{Z}.\}$ means all the linear combinations with integer coefficients. This $N-1$-dimensional affine space represents all the solutions of Bézout's identity made on the coordinates of **p**.

5. Conclusion

The problem treated in the present paper called the 'hyperplane unit-cell problem' consists of finding, for any hyperplane **p** of N dimensions, one short vector $\mathbf{b}_1$ that is such that $\mathbf{p}^t\mathbf{b}_1 = 1$ and $N-1$ short integer vectors $\mathbf{b}_2, \ldots, \mathbf{b}_j, \ldots, \mathbf{b}_N$ that are such that $\mathbf{p}^t\mathbf{b}_j = 0$. The short out-of-plane vector $\mathbf{b}_1$ is the solution of Bezout's identity with **p**, and the short in-plane vectors $\mathbf{b}_j$, $j \geq 2$, are solutions of the integer relation with **p**. These vectors constitute a unit cell attached to the hyperplane **p**. The algorithm to find a short solution to the N-dimensional Bézout's identity is presented in Section 2. The algorithm to find a solution to a connected problem called the column-constrained unimodular matrix (CCUM) is detailed in Section 3. Both algorithms are then combined with the help of an oblique projection to determine a small unit cell attached to any hyperplane **p** (Section 4). The vectors $\mathbf{b}_1, \ldots, \mathbf{b}_N$ are short and can be further shortened by lattice reduction. We have shown in some examples that the solutions of the integer relation are even shorter than those determined by the PSLQ algorithm computed with *Mathematica*.

APPENDIX A
Notations, Bézout's identity and integer relations

We note u_i the ith coordinate of a vector **u**. Sometimes, the notation $\mathbf{u}_{(i)}$ will be equivalently used. It should not be confused with $\mathbf{u}_i$ which is the ith vector in a set of vectors $\{\mathbf{u}_i\}$. The coordinates of a vector **u** are written in columns and those of a vector $\mathbf{u}^t$ are in rows. From a crystallographic point of view, column and row vectors belong to direct and reciprocal spaces, respectively. The matrix multiplication notation is adopted. It means that even a 'simple' scalar (inner) product $\mathbf{p} \cdot \mathbf{u} = \sum_i p_i u_i$ is written $\mathbf{p}^t\mathbf{u}$ where $\mathbf{p}^t$ means transpose of **p**.

Bézout's identity in 2D is an arithmetic theorem that states that for a and b which are integers with d for greatest common

Table 1
Equivalence of mathematic/crystallographic terms.

Mathematics	*Crystallography*
Bézout's identity: given an integer vector $\mathbf{p}$, find an integer vector $\mathbf{b}_1$ such that that $\mathbf{p}^t\mathbf{b}_1 = 1$	Given a plane $\mathbf{p}$, find a lattice vector $\mathbf{b}_1$ that points to a node of the first layer of $\mathbf{p}$. The vector $\mathbf{b}_1$ represents a translation between the layer $q = 0$ and $q = 1$
Integer relation: given an integer vector $\mathbf{p}$, find $N-1$ integer vectors $\mathbf{b}_j$ such that that $\mathbf{p}^t\mathbf{b}_j = 0$	Given a plane $\mathbf{p}$, find $N-1$ lattice vectors $\mathbf{b}_j$ that lie in the layer $q = 0$ of the plane $\mathbf{p}$
Set of solutions of Bézout's identity $\mathbf{b}_1 + \{\mathbb{Z}.\ \mathbf{b}_j\}$ with $j \in \{2, \ldots N\}$	The lattice unit cell made of vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_N$
Lattice reduction of the unit cell attached to $\mathbf{p}$: find the vectors $\mathbf{b}'_1$ and $\mathbf{b}'_j$ as short as possible and such that $\mathbf{p}^t\mathbf{b}'_1 = 1$ and $\mathbf{p}^t\mathbf{b}'_j = 0$	Lattice reduction of the unit cell attached to $\mathbf{p}$: find a unit cell such that the vector pointing to the first layer and the in-plane vectors are as short as possible

divisor, there exist integers u and v such that $au + bv = d$. More generally, the linear combinations of the form $au + bv$ are exactly the multiples of d. It can be generalized to any dimension N and written as follows. For any integer vector $\mathbf{p}$, calling d the greatest common divisor of the coordinates of $\mathbf{p}$, there exist integer vectors $\mathbf{u}$ such that $\mathbf{p}^t\mathbf{u} = \sum_i p_i u_i = d$. In the paper, we suppose that the coordinates of $\mathbf{p}$ are coprime, *i.e.* $d = 1$.

An integer relation between a real vector $\mathbf{x}$ exists if and only if there is an integer vector $\mathbf{u}$ such that $\mathbf{x}^t\mathbf{u} = \sum_i x_i u_i = 0$. There are different algorithms to determine integer relations, such as the PSLQ (Ferguson *et al.*, 1999). Searching for an integer relation between a set of powers of x $\{1, x, x_2, \ldots, x_n\}$ permits one to determine whether a given real number x is likely to be algebraic. Integer relations are also searched between some mathematical constants such as e, π and ln(2) in order to establish new arithmetic conjectures. In the paper, only the cases where $\mathbf{x}$ is an integer vector (called $\mathbf{p}$) are studied. (The equivalences between the mathematical and crystallographic terms used in the paper are given in Table 1.)

Acknowledgements

Professor Roland Logé is warmly acknowledged for the freedom given to our research that sometimes goes beyond metallurgy.

References

Bailey, D. H., Borwein, P. B. & Plouffe, S. (1997). *Math. C.* **66**, 903–914.
Capparelli, S. (2020). *Notes on Discrete Math*. Bologna, Italy: Società Editrice Esculapio.
Cayron, C. (2021). arXiv:2101.04500.
Ferguson, H. R. P., Bailey, D. H. & Arno, S. (1999). *Math. C.* **68**, 351–370.
Gorfman, S. (2020). *Acta Cryst.* A**76**, 713–718.
Lenstra, A. K., Lenstra, H. W. Jr. & Lovász, L. (1982). *Math. Ann.* **261**, 515–534.
Raayoni, G., Gottlieb, S., Manor, Y., Pisha, G., Harris, Y., Mendlovic, U., Haviv, D., Hadad, Y. & Kaminer, I. (2021). *Nature*, **590**, 67–73.
Wikipedia (2021*a*). https://en.wikipedia.org/wiki/Integer_relation_algorithm, accessed on 15th January 2021.
Wikipedia (2021*b*). https://w.wiki/znx, accessed on 15th January 2021.
Wikipedia (2021*c*). https://en.wikipedia.org/wiki/Coprime_integers, accessed on 15th January 2021.