

# Periodic Polyhedra

Benoît Meister

LSIIT - ICPS

Pôle API

Bd Sebastian Brant

F-67400 Illkirch

meister@icps.u-strasbg.fr

home page: <http://icps.u-strasbg.fr/~meister>

**Abstract.** This paper presents a new method for computing the integer hull of a parameterized rational polyhedron by introducing the concept of periodic polyhedron. Besides concerning generally parametric combinatorial optimization, the method has many applications for the analysis, optimization and parallelization of loop nests, especially in compilers.

## 1 Motivation

For many years, compiler writers have focused on parameterized loop nests, mainly because of their importance in scientific and multimedia programs. The *polytope model* [2] allows to manipulate loop nests whose bounds are affine functions with integer-valued parameters in the constant part by modeling them as parameterized rational polytopes  $P$ . As the considered loop indices are incremented by a constant integer value, the values taken by the  $n$ -vector of indices belong to a subset of  $\mathbb{Z}^n$ : an integer lattice  $L$ . So the values taken by the index vector  $I \in \mathbb{Z}^n$ , where  $n$  defines the number of nested loops, are given by the so-called  $\mathbb{Z}$ -polytope  $P \cap L$ .

*Example 1.* The iterations of the following Gaussian elimination code:

```
for(i=1; i<=n; i++)
  for(j=i+1; j<=n; j++)
    for(k=i+1; k<=n; k++)
      a[j][k]=a[j][k]-a[j][i]*a[i][k]/a[i][i];
```

are modeled by the parameterized  $\mathbb{Z}$ -polytope  $P \cap \mathbb{Z}^3$ , where

$$P = \{1 \leq i \leq n; i+1 \leq j \leq n; i+1 \leq k \leq n\}$$

and  $n$  is an integer parameter. □

As in the general problem of integer linear programming, we are interested in *integer points* (i.e., points with integer coordinates) in polytopes whose vertices may be non-integer.

In a class of code optimization and parallelization methods (e.g. in [20, 12, 21, 8, 22]), loop nests are usually transformed by applying an affine integer transformation to the  $\mathbb{Z}$ -polytope representing the loop nest [17]. They may also be split into sub-polyhedra as for example in [11]. The

result can then be transformed back into a loop nest by source code generation.

Several loop optimization and parallelization techniques also need to compute the indices of the first and last iterations to be executed in a loop nest (i.e. the lexicographic extrema of the  $I$  values). For instance, a precise dependence analysis may consist in computing the first executed iteration accessing a variable, among the set of iterations following a given access to this variable. Another issue is precise liveness analysis of data accessed in a loop nest, where the first and last iterations accessing the considered data have to be computed. Liveness analysis can be used to reduce the maximum amount of memory used by a program, and to reduce communications while parallelizing loops.

These techniques, as well as integer programming techniques in general, look for vertices of the *integer hull*  $P'$  of a parameterized  $\mathbb{Z}$ -polytope, as it is known that the solution is one of these vertices. The integer hull is the convex hull of the integer points in  $P \cap L$ .

Many techniques exist in the literature for finding integer vertices. Following Nemhauser and Wolsey ([9] chap. II) and Aardal *et al* [1], they can be classified as follows:

- *cutting plane* algorithms compute the integer hull of a rational polyhedron by adding constraints that cut the non-integer vertices out of it, so that all its integral points still respect the added constraint. The performance of this method is strongly related to the choice of a new cutting plane. The original idea has been proposed by Gomory in [10]. Seminal algorithms and complexity studies are due to Chvátal [3] and Schrijver [18].
- *branch and bound* algorithms recursively divide  $P$  into sub-polyhedra of which integer vertices are computed. The most commonly used technique partitions  $P$  into two polyhedra by a hyperplane including the optimal rational vertex of  $P$ .
- algorithms based on Lagrangean relaxation give an approximate solution by integrating constraints into the objective function.
- other algorithms that iteratively find vertices of the integer hull (for example: column generation).

Note that these methods usually do not compute the integer hull of  $P$  but only some of its vertices. Moreover, since we are interested in exact solutions, the Lagrangean relaxation is not suitable.

In this paper, we present a new way to compute the integer hull  $P'$  of a rational parameterized polytope  $P$ , defined by a set of rational affine equalities and inequalities on a set of variables  $I \in \mathbb{Z}^n$  and integer-valued parameters  $N \in \mathbb{Z}^p$ . Integer vertices are computed by recursively extracting, from the facets of  $P$ , sets of points having one more integer coordinate at each step.

Section 2 gives an intuitive explanation of the method. Section 3 then presents the class of mathematical objects to be used in this paper. The problem is then turned into a geometric form in section 4, which is used to devise an algorithm with some restrictions in section 5. These restrictions are eliminated in section 6. Finally, application of our method to find the integer lexicographic maximum is presented in section 7, while section 8 gives concluding remarks and future works.

## 2 Background and Principle

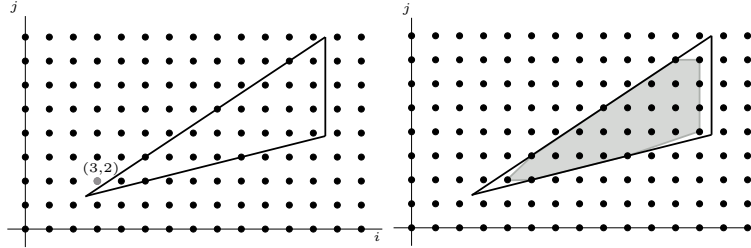
We assume that the considered polytope  $P$  is defined by a set of non-redundant constraints. This is not a strong assumption, since any polyhedron can be defined by such constraints (see [18] chap. 8), which can eventually be computed [19]. Without loss of generality, we consider  $\mathbb{Z}$ -polyhedra whose supporting lattice is  $\mathbb{Z}^n$ .

Since  $P$  is parameterized, the shape of  $P$  depends on the value of the parameters. Wilde and Loechner have shown in [14, 13] that the parameters space is partitioned into adjacent rational polyhedra, called *validity domains*, in which  $P$  has a given shape. Therefore, in a given validity domain, the coordinates of the vertices of  $P$  have a given affine expression. For simplicity, we first consider a polytope  $P$  whose parameters values belong to a unique validity domain. We show how to deal with the general case in section 6. Recall that we are concerned with integer-valued parameters. To have a better insight of the problem, let us consider the polytope presented in example 2.

*Example 2.* The following polytope is represented on figure 1.

$$P_2 = \{2i - 3j - 1 \geq 0; -i + 4j - 3 \geq 0; -2i + 25 \geq 0\}$$

□



**Fig. 1.**  $P_2$ , the *false* lexicographic minimum, and  $\text{int}(P_2)$

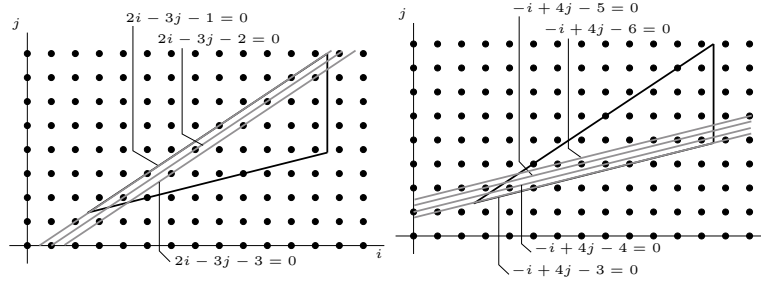
Its corresponding loop nest in C, generated by using the Fourier-Moztskin algorithm, would be:

```
for (i=ceil(13/5); i<= floor(25/2); i++)
  for (j=ceil((i+3)/4); j<=floor((2i-1)/3); j++)
    /* statements*/
```

According to the loop bounds, the minimal integer value for  $i$  is  $\lceil 13/5 \rceil = 3$ . The minimal integer value for  $j$  seems then to be  $\lceil (i+3)/4 \rceil = \lceil 6/4 \rceil = 2$ . But figure 1 shows that point  $(3, 2)$  does not belong to  $P_2$ : it does not correspond to a loop iteration. This example points out one of the problems one may encounter when manipulating polyhedra with non-integer vertices while dealing with integer points. This problem vanishes

if we manipulate the integer hull  $\text{int}(P_2)$  of  $P_2$  (in gray on figure 1), as its vertices are integer while it includes all the integer points (iterations) of  $P_2$ .

Let us now consider the maximum integer value of  $j$  for a given  $i$  in  $P_2$ . As one can see on figure 2, when  $i \bmod 3 = 0$ , the maximal integer



**Fig. 2.** integer minima and maxima for  $j$  for a given value of  $i$

value  $j_{max}$  is solution to the equation  $2i - 3j - 3 = 0$ . It is solution to  $2i - 3j - 1 = 0$  when  $i \bmod 3 = 2$ , and solution to  $2i - 3j - 2 = 0$  when  $i \bmod 3 = 1$ . Similarly, the minimal integer value  $j_{min}$  is solution to

$$\begin{cases} -i + 4j - 4 = 0 & \text{if } i \bmod 4 = 0 \\ -i + 4j - 3 = 0 & \text{if } i \bmod 4 = 1 \\ -i + 4j - 6 = 0 & \text{if } i \bmod 4 = 2 \\ -i + 4j - 5 = 0 & \text{if } i \bmod 4 = 3 \end{cases}$$

For a given value of  $i$ , the coordinate  $j$  of an integer point of  $P$  is bounded by values which are solutions to some equations. These equations depend periodically on  $i$ . So each integer point  $(i, j)$  of  $P_2$  is in the convex hull of two integer *extremal* points:  $(i, j_{min})$  and  $(i, j_{max})$ . The convex hull of all these extremal points is then the integer hull of  $P_2$ .

To generalize this idea, consider that any integer point

$$X = (x_1, x_2, \dots, x_n)$$

of  $P$  belongs to the set of points  $d_k(X)$  of  $P$  defined by

$$d_k(X) = \{I = (x_1, \dots, x_{k-1}, i_k, x_{k+1}, \dots, x_n), i_k \in \mathbb{Q}, I \in P\}, k \in [1..n]$$

$i_k$  has a minimal and maximal value in  $d_k(X)$ , given by two faces (or constraints) of  $P$ :  $p_{min}(I, N) \geq 0$  and  $p_{max}(I, N) \geq 0$ . Coefficient  $a_k$  of  $i_k$  in  $p_{min}(I, N)$  is positive and it is negative in  $p_{max}(I, N)$ , therefore we have:  $i_k \geq p'_{min}(I, N)$  and  $i_k \leq p'_{max}(I, N)$ . These inequalities define the *rational* lower and upper bounds for  $i_k$  in  $d_k(X)$ . In other terms, we have  $p'_{min}(I, N) \leq x_k \leq p'_{max}(I, N)$ . Furthermore, as  $X_k$  is integer, we have  $\lceil p'_{min}(X, N) \rceil \leq x_k \leq \lfloor p'_{max}(X, N) \rfloor$ . So  $X$  belongs to the convex hull of the integer points  $x_{min}(X, N)$  and  $x_{max}(X, N)$ , where

$$x_{min}(X, N) = (x_1, \dots, x_{k-1}, \lceil p'_{min}(X, N) \rceil, x_{k+1}, \dots, x_n)$$

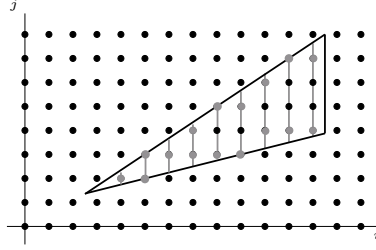
and

$$x_{max}(X, N) = (x_1, \dots, x_{k-1}, \lfloor p'_{max}(X, N) \rfloor, x_{k+1}, \dots, x_n).$$

It follows that:

- the integer hull  $int(P)$  of  $P$  is the convex hull of all the existing integer points  $x_{min}(X, N)$  and  $x_{max}(X, N)$ ,
- the vertices of  $int(P)$  are some of these points.

*Example 3.* The existing points  $x_{min}(i)$  and  $x_{max}(i)$  are represented on figure 3, as well as the integer hull of  $P_2$  and the different  $d_2(i, j)$  (with  $i = i_1$  and  $j = i_2$ ). One can see that the convex hull of all these points is the integer hull of  $P$ .  $\square$



**Fig. 3.** The extremal integer points of  $P_2$  and the different  $d_2(i, j)$

We have seen that the extremal points are periodically solution to an equality. We explain this periodic character in next section.

### 3 Periodic Polyhedra

Consider the  $k^{th}$  coordinate of  $x_{max}$ , obtained from the inequality

$$p_{max}(I, N) = \sum_{t=1}^n a_t i_t + \sum_{t=1}^n b_t n_t + c \geq 0 :$$

$$x_{max,k} = \lfloor p'_{max}(X, N) \rfloor = \lfloor \sum_{t=1}^n a'_t x_t + \sum_{t=1}^p b'_t n_t + c' \rfloor$$

where  $a'_k = 0, a'_t = a_t/(-a_k), b'_t = b_t/(-a_k)$  and  $c' = c/(-a_k)$  ( $a_k$  is negative). The common denominator of the rational coefficients of  $p'_{max}(X, N)$  is  $-a_k$ .

Let  $a$  be an integer and  $b$  a positive integer. We have  $\lfloor \frac{a}{b} \rfloor = \frac{a}{b} - \frac{a \bmod b}{b}$

Let  $p''_{max} = -a_k \cdot p'_{max}$ . Then we can write:  $x_{max,k} = \lfloor \frac{p''_{max}(X, N)}{-a_k} \rfloor$

$$x_{max,k} = \frac{p''_{max}(X, N)}{-a_k} - \frac{p''_{max}(X, N) \bmod (-a_k)}{-a_k}$$

$$-a_k x_{max,k} = p_{max}''(X, N) - p_{max}'(X, N) \bmod (-a_k)$$

As  $p_{max}'(X, N) + a_k x_k = p_{max}(X, N)$ , this reduces to:

$$p_{max}(X, N) - p_{max}(X, N) \bmod (-a_k) = 0 \Leftrightarrow x_k = x_{max,k} \quad (1)$$

Similarly, we have:

$$p_{min}(X, N) - p_{min}(X, N) \bmod (a_k) = 0 \Leftrightarrow x_k = x_{min,k} \quad (2)$$

Equations (1) and (2) can be summed up into Theorem 1:

**Theorem 1.** *Let  $p(X, N) \geq 0$  be an inequality with integer coefficients. The value of  $x_k$  that is solution to*

$$\mathcal{I}(p(X, N), i_k) = 0, \quad (3)$$

*for integer values of  $(x_1, x_2, \dots, x_{k-1}, x_{k+1}, \dots, x_n)$ , is the extremal integer value for  $x_k$  given by  $p(X, N) \geq 0$ , where  $\mathcal{I}(p(X, N), i_k) = p(X, N) - p(X, N) \bmod (|a_k|)$ .*

So the set of extremal points  $x_{min}(X, N)$  and  $x_{max}(X, N)$  are integer solutions to (3) for some constraint  $p(X, N) \geq 0$  of  $P$ . In [16], we have shown that if  $f(I)$  is an affine function with integer coefficients and  $m$  is integer,  $f(I) \bmod m$  is an integer *periodic number*, with  $I \in \mathbb{Z}^n$ . A periodic number is a rational-valued periodic function of  $I$  (see [6, 4]). It can be represented by a  $n$ -dimensional array whose number of elements in the  $k^{th}$  dimension is the corresponding period  $s_k$ .

*Example 4.* A 2-dimensional periodic number of period  $S = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$  de-

pending on  $(N, M) \in \mathbb{Z}^2$ , whose value is :

- 1 for  $N \bmod 2 = 0, M \bmod 3 = 0$
- 2 for  $N \bmod 2 = 1, M \bmod 3 = 0$
- 3 for  $N \bmod 2 = 0, M \bmod 3 = 1$
- 4 for  $N \bmod 2 = 1, M \bmod 3 = 1$
- 5 for  $N \bmod 2 = 0, M \bmod 3 = 2$
- 0 for  $N \bmod 2 = 1, M \bmod 3 = 2$

can be represented as:  $\begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 0 \end{bmatrix}_{N,M}$  □

Consider  $A(i) = i \bmod 3, i \in \mathbb{Z}$ . Its value is periodically 0, 1, and 2, depending on  $i$ . Hence it is a periodic number of period 3. In a more general way, if  $f(I) = \sum_{k=1}^n a_k i_k + c$  is an integer affine function of  $I \in \mathbb{Z}^n$  with  $a_k, c \in \mathbb{Z}$ , and  $m \in \mathbb{N}$ ,  $f(I) \bmod m$  is a periodic integer number of period  $S = (s_k)$  with  $s_k = \frac{m}{\gcd(m, |a_k|)}$ .

*Example 5.* With  $f(I) = 3i + 4j$ , and  $m = 6$ , it gives:

$$(3i + 4j) \bmod 6 = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \end{bmatrix}_{(3i+4j)} = \begin{bmatrix} 0 & 4 & 2 \\ 3 & 1 & 5 \end{bmatrix}_{i,j} \quad \square$$

We can then give an explicit form of the equalities yielding the extremal points by using periodic numbers.

*Example 6.* According to theorem 1, the  $j$  coordinate of the extremal points of  $P_2$  is given by one of the two equations:

$$-i + 4j - 4 - (-i + 4j - 3) \bmod 4 = -i + 4j - 3 - [1 \ 0 \ 3 \ 2]_i = 0$$

$$2i - 3j - 1 - (2i - 3j - 1) \bmod 2 = 2i - 3j - 1 - [1 \ 0]_i = 0,$$

which is concordant with our preliminary observations.  $\square$

These equalities depend on a periodic number: their *definition* is periodic. As a classical equality defines a hyperplane, i.e. a polyhedron, an equality whose definition is periodic defines then a periodic polyhedron. In [16], we present a general class of mathematical objects depending periodically on multiple parameters, which are called *periodics*.

**Definition 1.** A  $n$ -dimensional periodic polyhedron  $M$  of period  $S = (s_1 \ s_2 \ \dots \ s_n)^T \in \mathbb{N}^n$  is given by:

–  $q = s_1 \times s_2 \times \dots \times s_n$  polyhedra  $M_I$ , indexed by:

$$I = (i_1 \ i_2 \ \dots \ i_n)^T \in \mathbb{Z}^n \text{ with } 0 \leq i_k < s_k, k \in [1..n]$$

– their respective definition domain  $\mathcal{D}_I$  : the integer lattice defined by:

$$\begin{pmatrix} s_1 & 0 & \dots & 0 \\ 0 & s_2 & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \dots & s_m \end{pmatrix} J + I, J \in \mathbb{Z}^n$$

$\bigcup_I \mathcal{D}_I = \mathbb{Z}^n$ : any element  $X \in \mathbb{Z}^n$  is mapped to a unique  $M_I$  by the relation  $I = X \bmod S$ .  $\square$

Notice that each  $M_I$  is then the intersection between an integer lattice and a polyhedron (i.e. a  $\mathbb{Z}$ -polyhedron).

*Example 7.* A 2-dimensional periodic over polyhedra (made of one inequality) of  $(i, j, k) \in \mathbb{Z}^3$  depending on  $(i, j) \in \mathbb{Z}^2$ , whose value is :

- $i + 2j - 3k + 7 \geq 0$  for  $i \bmod 3 = 0, j \bmod 2 = 0$
- $2i - j + k \geq 0$  for  $i \bmod 3 = 1, j \bmod 2 = 0$
- $3j - 5 \geq 0$  for  $i \bmod 3 = 2, j \bmod 2 = 0$
- $-i - 2j \geq 0 + 3k - 7$  for  $i \bmod 3 = 0, j \bmod 2 = 1$
- $i - 2k + 1 \geq 0$  for  $i \bmod 3 = 1, j \bmod 2 = 1$
- $1 \geq 0$  for  $i \bmod 3 = 2, j \bmod 2 = 1$

can be represented as:

$$\begin{bmatrix} i + 2j - 3k + 7 \geq 0 & -i - 2j + 3k - 7 \geq 0 \\ 2i - j + k \geq 0 & i - 2k + 1 \geq 0 \\ 3j - 5 \geq 0 & 1 \geq 0 \end{bmatrix}_{i,j}, \text{ and can also be written:}$$

$$\begin{bmatrix} 1 & -1 \\ 2 & 1 \\ 0 & 0 \end{bmatrix}_{i,j} i + \begin{bmatrix} 2 & -2 \\ -1 & 0 \\ 3 & 0 \end{bmatrix}_{i,j} j + \begin{bmatrix} -3 & 3 \\ 1 & -2 \\ 0 & 0 \end{bmatrix}_{i,j} k + \begin{bmatrix} 7 & -7 \\ 0 & 1 \\ -5 & 1 \end{bmatrix} \geq 0$$

(we call this form *factorized form*, as the constant form of an affine function of  $(i, j, k)$  is extracted).  $\square$

Using the knowledge of periodic polyhedra, we show in the next section new geometric tools that will allow to give an algorithm for computing the integer hull of a polyhedron.

## 4 Pseudo-facets

We have seen that extremal points of  $P$  are integer solutions to an equality  $\mathcal{I}(p(X, N), i_k) = 0$  where  $p(X, N) \geq 0$  is one of the inequalities that define  $P$ . As well as the inequality  $p(X, N) \geq 0$  defines a facet of  $P$  (with its adjacent faces), we say that the solutions to  $\mathcal{I}(p(X, N), i_k) = 0$  belong to a *pseudo-facet* of  $P$ .

**Definition 2.** The  $q^{th}$  facet  $f_q(P)$  of a polyhedron  $P$  can be defined by:  $\{I \in P \mid p_q(I, N) = 0\}$ . Similarly, the  $q^{th}$  pseudo-facet  $f'_{q,k}(P)$  of  $P$  w.r.t.  $i_k$  is defined by:  $\{I \in P \mid \mathcal{I}(p_q(I, N), i_k) = 0\}$ .

A pseudo-facet  $f_q(P)$  can be decomposed into:

- a (supporting) pseudo-hyperplane, defined by  $\mathcal{I}(p_q(I, N), i_k) = 0$ ,
- a projected pseudo-facet, which is the projection of the pseudo-facet along  $i_k$  by using the equality  $\mathcal{I}(p_q(I, N), i_k) = 0$ .

The projected pseudo-facet defines the values of all the variables but  $i_k$  for which the solution of  $\mathcal{I}(p_q(I, N), i_k) = 0$  belongs to the pseudo-facet  $f'_{q,k}(P)$ .

*Example 8.* Consider the parameterized polyhedron  $P_3$  defined by:

$$P_3 = \{-2i_1 + 3i_2 - n \geq 0; -2i_2 + 21 \geq 0; 4i_1 + i_2 - 13 \geq 0\}$$

Figure 8 shows the facet of  $P$  defined by :

$$\begin{cases} -2i_1 + 3i_2 - n = 0 \\ -2i_2 + 21 \geq 0 \\ 4i_1 + i_2 - 13 \geq 0 \end{cases} = \begin{cases} -2i_1 + 3i_2 - n = 0 \\ 2i_2 + 21 \geq 0 \\ 7i_2 - 2n - 13 \geq 0 \end{cases}$$

and the corresponding pseudo-facet w.r.t.  $i_1$ :

$$\begin{aligned} & \begin{cases} \mathcal{I}(-2i_1 + 3i_2 - n, i_1) = -2i_1 + 3i_2 - n - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}_{i_2, n} = 0 \\ -2i_2 + 21 \geq 0 \\ 4i_1 + i_2 - 13 \geq 0 \end{cases} \\ &= \begin{cases} -2i_1 + 3i_2 - n - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}_{i_2, n} = 0 \\ \{-2i_2 + 21 \geq 0; 7i_2 - 2n - \begin{bmatrix} 13 & 15 \\ 15 & 13 \end{bmatrix}_{i_2, n} \geq 0\} \end{cases} \end{aligned}$$

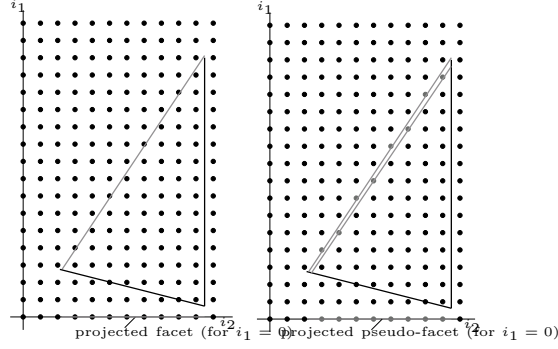
□

This allows a more geometrical expression of the theorem 1, as the integer points whose convex hull is the integer hull of  $P$  belong to the pseudo-facets of  $P$  w.r.t.  $i_k$ .

**Theorem 2.** The convex hull of the integer hulls of the pseudo-facets of  $P$  w.r.t.  $i_k$  is the integer hull of  $P$ .

*Proof.* Any (integer) extremal point belongs to the integer hull of a pseudo-facet of  $P$  w.r.t.  $i_k$ . The convex hull of all the existing integer hulls is then the integer hull of the extremal points, that is to say the integer hull of  $P$ .





**Fig. 4.** A facet of  $P_3$  and its corresponding pseudo-facet w.r.t.  $i_1$ .

The determination of the integer hull of  $P$  therefore requires to compute the integer hull of each of the pseudo-facets of  $P$  w.r.t.  $i_k$ . But a pseudo-facet is a periodic polyhedron. We have shown in [15] that the pseudo-facet of a periodic polyhedron  $P'$  w.r.t. variable  $i_k$  is computed by taking into account the period of  $P$  along  $i_k$ . The supporting hyperplane is given by:

$$\mathcal{I}(\sum_l a_l i_l + c, i_k) = \sum_l a_l i_l + c - ((\sum_l a_l i_l + c) \bmod |a_k s_k|),$$

where  $s_k$  is the period of  $P'$  along  $i_k$ . Computing pseudo-facets for pseudo-polyhedron allows us to compute the integer hull of a pseudo-facet. This is used by the algorithm for computing the integer hull of a parameterized rational polyhedron, as presented in the next section.

## 5 integer hull

Theorem 2 states a relation between the integer hull  $int(P)$  of polyhedron  $P$  and the integer hull of its pseudo-facets w.r.t. variable  $i_k$ :

$$int(P) = conv(\bigcup_q int(f'_{q,k}(P)))$$

The problem is now to compute  $int((f'_{q,k}(P)))$ . Similarly to the integer hull of  $P$ , the integer hull of  $f'_{q,k}$  is the integer hull of its pseudo-facets w.r.t. another variable  $i_{k'}, k' \in [1..n] \setminus k$  as the points of  $f'_{q,k}$  correspond already to an integer bound of  $i_k$ . Thus we have:

$$int((f'_{q,k})) = conv(\bigcup_{q'} int(f''_{q',k'}(f'_{q,k}))) \quad (4)$$

This establishes a recurrence relation between the integer hull of a pseudo-facet and the integer hull of its own pseudo-facets. According to theorem 1, a property of a pseudo-facet w.r.t.  $i_k$  is that  $i_k$  is integer if the other variables and the parameters are integer. A pseudo-facet  $f''$

w.r.t  $i_{k'}$  of a pseudo-facet  $f'$  w.r.t.  $i_k$  belongs to  $f'$ :  $i_k$  is integer if the other variables and the parameters are integer. Moreover, it is a pseudo-facet w.r.t.  $i_{k'}$ , so  $i_{k'}$  is integer if the other variables and the parameters are integer. Then  $i_k$  and  $i_{k'}$  are integer if the other variables are integer. Recursively, taking  $n$  times a pseudo-facet of a pseudo-facet leads to a pseudo-facet  $f$  such that all the variables are integer when the parameters are integer. Since a pseudo-facet of a periodic polyhedron of dimension  $m$  is  $(m-1)$ -dimensional,  $f$  is of dimension 0. The two latter sentences state that  $f$  is an integer point. The integer hull of  $P$  is then the convex hull of all the  $f$ 's. The relations among  $P$ , its pseudo-facets and recursively the pseudo-facets of the pseudo-facets are given by a tree, the *pseudo-facet tree*  $P$ . The vertices of  $\text{int}(P)$  are obtained by scanning this tree from its root  $P$  to its 0-dimensional pseudo-facets (the vertices), called the *pseudo-vertices*. We can devise an algorithm from the recurrence relation among  $P$  and its pseudo-facets to obtain the pseudo-vertices of  $P$ .

```

get_pseudo_vertices(periodic polyhedron P) {
  n = dimension(P)
  if n = 0 return P
  U = empty set of periodic polyhedra
  k = rank of the variable to be processed for n
  i = the kth variable of V
  for each inequality f(I) >= 0 of P
    with a nonzero coefficient for i do :
      compute P' by replacing f(I) >= 0 in P
        by f(I) - (f(I) mod |a|) = 0
        ==> P' = {P | f(I) - (f(I) mod |a|) = 0}
      P_proj = P' projected pseudo-facet of P'
      add get_pseudo_vertices(P_proj) to U
      for each element u of U
        add ({ f(I) - (f(I) mod |a|) = 0 } ∩ u) to P''
      endfor
    endfor
  return P''
}

```

By opposition to cutting planes algorithms, our algorithm adds no constraint to the original problem, but constraints are replaced by periodic constraints. It adds neither variables nor parameters to the problem. Moreover, the pseudo-facet tree of  $P$  is  $n$ -deep, so a given pseudo-vertex is obtained in computing a pseudo-facet  $n$  times.

However, it handles periodic numbers, which can have large periods if the coefficients of the constraints of  $P$  are large. An upper bound of the number of elements of the computed periodics is  $M^{n+p}$ , where  $M$  is the maximal value of a coefficient that can be obtained by row elimination of the constraints of  $P$ , and  $n$  and  $p$  are the number of variables and parameters. One can still consider the use of the symbolic expression of a periodic number, especially when  $M$  is large, to achieve the computation. Computing the pseudo-vertices of a  $n$ -dimensional simplex requires  $(n+1)!$  computations of a pseudo-facet, which gives a lower complexity bound. Notice that the algorithm is highly parallel: the processing of subtrees can be distributed, and the only data to be communicated is the pseudo-facet corresponding to the subtree root.

## 6 Generalization

### 6.1 Unbounded polyhedra

Unbounded polyhedra can be used to model nested *while* and *for* loops with the polytope model, as for instance in [5]. Nemhauser and Wolsey have proved in [9](part I, section 6) that the integer hull  $S$  of an unbounded polyhedron  $P$  defined by its Minkowsky representation:

$$P = \{x \in \mathbb{R}^n | x = \sum_k \lambda_k v^k + \sum_j \mu_j r^j, \sum_k \lambda_k = 1, \mu_j \geq 0 \forall j\}$$

where the  $v^k \in \mathbb{R}^n$  are the extreme vertices of  $P$  and  $r^j$  its extreme rays, is given by:

$$S = \{x \in \mathbb{Z}^n | x = \sum_k \lambda_k q^k + \sum_j \mu_j r^j, \sum_k \lambda_k = 1, \mu_j \geq 0 \forall j\}$$

where the  $q^k$  are the extremal vertices of the integer hull of  $\text{conv}(v^k)$ . Thus, computing the integer hull of an unbounded polyhedron  $P$  amounts to compute the integer hull of its extreme vertices.

### 6.2 Polyhedron with multiple validity domains

In the general case of a parameterized polyhedron, the computed pseudo-vertices will actually belong to the polyhedron for only some values of the parameters. Loechner has given in [13] an algorithm to compute the values of parameters for which a given rational vertex belongs to a rational polyhedron. The parameters space is then partitioned into polyhedral domains in which a given set of vertices belong to  $P$ . This method can be adapted directly to compute pseudo-vertices: we can partition the parameter space into periodic polyhedra in which a given set of pseudo-vertices belong to  $P$ . This is translated into the following algorithm:

```

get_integer_hull(Polyhedron P) {
  IH = empty list of periodic polyhedra
  PX = empty list of periodic polyhedra
  PX = get_pseudo_vertices(P)
  (V, X) = lists of validity domains and
           the corresponding valid pseudo-vertices from PX
  for each (V_i, X_i) in (V, X)
    H_i = convex hull of the pseudo-vertices X_i
    add H_i to H
  endfor
  return (V, H)
}

```

### 6.3 Non-fully-dimensional polyhedra

The geometric dimension of a polyhedron  $P$  defined in a  $n$ -dimensional variable space by a non-redundant system of  $e$  equalities and  $e'$  inequalities is given by  $d(P) = n - e$ . A polyhedron is fully-dimensional if  $d(P) = n$ , so if  $P$  has no equalities. Integer solutions to a system of  $e$  equalities belong to a  $(n - e)$ -dimensional integer lattice, whose determinant may be greater than 1 and which depends on the parameters. Methods to deal with this case will be presented in a further work.

## 7 An application: computing the integer lexicographic extremum of a polytope

The lexicographic order is a total order on the elements of a  $n$ -dimensional space. It is noted  $\triangleleft$  and defined by:

$$I = (i_1 \ i_2 \ \dots \ i_n) \in \mathbb{Z}^n \triangleleft J = (j_1 \ j_2 \ \dots \ j_n) \in \mathbb{Z}^n \\ \Leftrightarrow \exists k \in [1..n] : i_1 = j_1, i_2 = j_2, \dots, i_{k-1} = j_{k-1}, i_k < j_k$$

For loop nests in computer programs, the integer lexicographic minimum (resp. maximum) in  $P \cap \mathbb{Z}^n$  corresponds to the first (resp. last) iteration to be executed. Feautrier presents in [7] a method to find the lexicographic extrema of a parameterized loop nest, based on parameterized versions of Gomory's fractional cutting planes approach and the lexicographic dual simplex algorithm.

Both integer lexicographic maximum or minimum are obtained in a similar way, roughly by inverting comparison and extrema operators. So we focus on the computation of the integer lexicographic maximum. The integer lexicographic maximum of  $P$  belongs to the set of integer points  $X_1$  of  $\mathbb{Z}^n \cap P$  having the maximal existing integer value for  $i_1$ . If such a set contains several points, the integer lexicographic maximum is in the subset of  $X_1$  having the maximal integer value for  $i_2$ , and so on. As for any solution to a linear integer programming problem, we know that the lexicographic maximum is a vertex of the integer hull of  $P$ .

A straightforward way to compute the integer lexicographic maximum of  $P$  would therefore be to compute its integer hull  $P'$  and then to find the lexicographic rational maximum in  $P'$ , for example by using the lexicographic dual simplex algorithm (see [7] or [9] chap. II section 4). But this approach yields the computation of every vertex, which is costly and in general useless as some vertices can never be the lexicographic maximum. So, as it is generally done in integer linear programming, we must compute only a subset of the vertices of  $P'$ , containing the integer lexicographic maximum of  $P$ . We can equivalently compute the set of vertices of  $P'$  that excludes a maximal number of vertices that cannot be the lexicographic maximum. The efficiency of the method depends on the selection of the pseudo-facets that will contain the candidate vertices. The maximal *rational* values of  $i_1$  are those of the faces of  $P$  which are upper bounds for  $i_1$ : those whose coefficient on  $i_1$  is negative. Candidate vertices therefore belong to the corresponding pseudo-facets  $f$  of  $P$  w.r.t.  $i_1$ , which eliminates the vertices that are in the other pseudo-facets. From now on, we will call *min-pseudo-facet* (resp. *max-pseudo-facet*) of a periodic polyhedron  $P$  w.r.t.  $i_k$  a pseudo-facet corresponding to an integer lower (resp. upper) bound of  $i_k$  in  $P$ .

*Example 9.* The lexicographic maximum of  $P_3$  belongs to its only max-pseudo-facet w.r.t.  $i_1$ , defined by:

$$f(I, N) = \left\{ \begin{array}{l} -2i_1 + 3i_2 - n - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}_{i_2, n} = 0 \\ \{-2i_2 + 21 \geq 0; 7i_2 - 2n - \begin{bmatrix} 13 & 15 \\ 15 & 13 \end{bmatrix}_{i_2, n} \geq 0\} \end{array} \right.$$

□

Let  $a_1 i_1 + a_2 i_2 + \dots a_n i_n + a_0 = 0$  be the associated supporting pseudo-hyperplane. For a given value of the parameters, maximizing  $i_1$  is equivalent to maximize  $g(I) = a_2 i_2 + \dots a_n i_n$ , since  $i_1 = -\frac{a_2 i_2 + \dots + a_n i_n}{a_1}$  and  $a_1 < 0$ . This is a second elimination criterion, which allows to keep the pseudo-facets of  $f$  that can contain the integer points with a maximal value for  $g(I)$ . Let us see two ways to select relevant pseudo-facets of  $f$ . First, we can consider the variation of  $g(I)$  w.r.t. the sign of  $a_k$ .  $g(I)$  increases with variables  $i_{inc}$  with  $a_k > 0$ . Thus, the candidate vertices belong to pseudo-facets that maximize such  $i_{inc}$ . But  $g(I)$  also decreases with variables  $i_{dec}$  with a negative coefficient and is constant for  $i_{cst}$  with a zero coefficient. In the same way, we will also consider pseudo-facets that minimize  $i_{dec}$ . Besides, we have to include max-pseudo-facets as well as min-pseudo-facets w.r.t.  $i_{cst}$ .

*Example 10.* In our example,  $i_2$  increases in function of  $i_1$ , so the lexicographic maximum belongs to the max-pseudo-facets of  $f(I, N)$  w.r.t.  $i_2$ :  $\{-2i_1 + 3i_2 - n - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}_{i_2, n} = 0; -2i_2 + \begin{bmatrix} 20 & 18 \end{bmatrix}_{i_2} = 0\}$ . The candidate pseudo-vertex is then:  $(i_1, i_2) = ((-n + \begin{bmatrix} 30 & 29 \\ 26 & 27 \end{bmatrix}_{i_2, n})/2, \begin{bmatrix} 10 & 9 \end{bmatrix}_{i_2})$ . For each value of  $n$ , the maximal value of  $i_1$  is reached for  $i_2 \bmod 2 = 0$ . The integer lexicographic maximum is then:  $(i_1, i_2) = (-n + \begin{bmatrix} 30 & 29 \end{bmatrix}_n)/2, 10)$  □

This pseudo-facet selection method can be written as an algorithm, detailed in [15].

In general, each pseudo-vertex depends periodically on the variables and the parameters. But there is always one optimal value of the variables for each value of the parameters, which must be selected.

*Example 11.* Let us make a sketch of the process to find the candidate vertices for a lexicographic maximum with the following example (also considered by Feautrier in [7]):  $P = \{-i + m \geq 0; i \geq 0; j \geq 0; -j + n \geq 0; 2i + j - k = 0\}$  where  $k, n, m$  are the parameters. Since there is an equality, for a given (maximal)  $i$ ,  $j$  is determined, so we can eliminate  $j$ . The coefficient in  $j$  is 1, so  $j$  is integer for any integer value of the variables and the parameters. Therefore, the problem presented in section 6.3 does not occur here.

$$P = \left\{ \begin{array}{l} 2i + j - k = 0 \\ \{-i + m \geq 0; i \geq 0; -2i + k \geq 0; 2i - k + n \geq 0\} \end{array} \right.$$

The candidate vertices belong to two max-pseudo-facets of  $P$  w.r.t.  $i$ :

$$A = \left\{ \begin{array}{l} 2i + j - k = 0 \\ \begin{cases} -i + m = 0 \\ i \geq 0 \\ -2i + k \geq 0 \\ 2i - k + n \geq 0 \end{cases} \end{array} \right. = \left\{ \begin{array}{l} 2i + j - k = 0 \\ \begin{cases} -i + m = 0 \\ m \geq 0 \\ k - 2m \geq 0 \\ -k + 2m + n \geq 0 \end{cases} \end{array} \right.$$

$$B = \left\{ \begin{cases} 2i + j - k = 0 \\ -i + m \geq 0 \\ i \geq 0 \\ -2i + k - (-2i + k) \bmod 2 = 0 \\ 2i - k + n \geq 0 \end{cases} \right\} = \left\{ \begin{cases} 2i + j - k = 0 \\ -2i + k - [0 \ 1]_k = 0 \\ -k + 2m + [0 \ 1]_k \geq 0 \\ k - [0 \ 1]_k \geq 0 \\ n - [0 \ 1]_k \geq 0 \end{cases} \right\}$$

Here,  $A$  and  $B$  are the pseudo-vertices we are looking for:

$$A : \begin{pmatrix} i \\ j \end{pmatrix} = \begin{pmatrix} m \\ k - 2m \end{pmatrix} \text{ for } \begin{cases} m \geq 0 \\ 2m \leq k \leq 2m + n \end{cases}$$

$$B : \begin{pmatrix} i \\ j \end{pmatrix} = \begin{pmatrix} \frac{k - [0 \ 1]_k}{2} \\ [0 \ 1]_k \end{pmatrix} \text{ for } \begin{cases} n \geq [0 \ 1]_k \\ [0 \ 1]_k \leq k \leq 2m + [0 \ 1]_k \end{cases}$$

□

In the general case, the domains of parameters are unions of polyhedra. As the algorithm computes extrema one variable at a time, the result can take the form of a quasi-affine syntax tree (as in Feautrier's PIP) or of a union of polyhedra.

Another pseudo-facet selection method would be to select the pseudo-facets corresponding to faces of equation  $\alpha_1 i_1 + \alpha_2 i_2 + \dots \alpha_n i_n + \alpha_0$  such that  $(0 \ a_2 \ \dots \ a_n) \cdot (\alpha_1 \ \alpha_2 \ \dots \ \alpha_n)^T \leq 0$ . Such faces are rational maximal bounds for  $g(I)$ , which is easily figured out by a rational linear transformation.

## 8 Conclusion

We have defined a new method for computing the integer hull of a rational parameterized polyhedron, which has many applications in loop nests analysis and optimization. This geometric method also yields new ways to solve parametric integer programming problems. In particular, it can help solving integer lexicographic extrema, which is an omnipresent problem in loop nest analysis, optimization and parallelization. One of our objectives was to provide a method based on the underlying concepts of the polyhedral library Polylib [19]. While considering this objective, we particularly generalized the concept of periodic numbers to periodic polyhedra. The presented algorithms are being implemented in the library. Their performance will be compared with the existing solvers.

Other problems can also be tackled using our method: the domain of parameters for which an integer point exists in a polyhedron, the number of integer points in the projection of a  $\mathbb{Z}$ -polyhedron, and alternative ways to compute Ehrhart polynomials are currently worked on. We also plan to write a generic parametric integer solver, by finding better pseudo-facet elimination criteria, and to compare our methods *in silico* with the existing ones.

## References

1. K. Aardal and S. van Hoesel. Polyhedral techniques in combinatorial optimization ii: Computations. *Statistica Neerlandica*, 50:3, 1996.

2. C. Ancourt and F. Irigoien. Scanning polyhedra with DO loops. In *Proc. ACM SIGPLAN '91*, pages 39–50, June 1991.
3. V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4:305–337, 1973.
4. Ph. Clauss. Counting solutions to linear and nonlinear constraints through Ehrhart polynomials: Applications to analyze and transform scientific programs. In *10th ACM Int. Conf. on Supercomputing, Philadelphia*, 1996.
5. J.-F. Collard. Space-time transformation of while-loops using speculative execution. In *proc. of the Scalable High-Performance Computing Conf. (SHPCC'94)*, pages 429–436. IEEE, May 1994.
6. E. Ehrhart. *Polynômes arithmétiques et Méthode des Polyèdres en Combinatoire*, volume 35 of *International Series of Numerical Mathematics*. Birkhäuser Verlag, Basel/Stuttgart, 1977.
7. P. Feautrier. Parametric integer programming. *RAIRO Recherche Opérationnelle*, 22:243–268, September 1988.
8. P. Feautrier. Lncs 1132. In *The Data Parallel Programming Model*, chapter Automatic parallelization in the Polytope Model, pages 79–100. 1996.
9. G. Nemhauser and L. Wolsey. *Integer and Combinatorial Optimization*. W-I Series in Discrete Mathematics and Optimization. Wiley-Interscience, 1988. ISBN 0-471-82819-X.
10. R. E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the AMS*, 64:275–278, 1958.
11. Martin Griebl, Paul A. Feautrier, and Christian Lengauer. Index set splitting. *Int. J. Parallel Programming*, 28(6):607–631, 2000.
12. C. Lengauer. Loop parallelization in the polytope model. In *CONCUR 93*, 1993.
13. V. Loechner. *Contribution à l'étude des polyèdres paramétrés et applications en parallélisation automatique*. PhD thesis, Université Louis Pasteur, 1997. <http://icps.u-strasbg.fr/pub-97/>.
14. V. Loechner and D. K. Wilde. Parameterized polyhedra and their vertices. *Int. J. of Parallel Programming*, 25(6):525–549, Dec. 1997.
15. B. Meister. Periodic polyhedra. research report, LSIIT - ICPS, October 2003.
16. B. Meister. Using periodics in integer polyhedral problems. research report, LSIIT - ICPS, 2003. <http://icps.u-strasbg.fr/~meister>.
17. P. Quinton, S. V. Rajopadhye, and T. Riset. On manipulating z-polyhedra using a canonical representation. *Parallel Processing Letter*, 7(2):181–194, 1997.
18. A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley and Sons, 1986.
19. D.K. Wilde. A library for doing polyhedral operations. Technical report 785, IRISA, Rennes, France, 1993.
20. M.E. Wolf and M.S. Lam. A loop transformation theory and an algorithm to maximize parallelism. *IEEE Transactions on Parallel and Distributed Systems*, 2(4):452–471, October 1991.
21. M. Wolfe. *High Performance Compilers for Parallel Computing*. Addison Wesley, 1996.
22. J. Xue. Unimodular transformations of non-perfectly nested loops. *Parallel Computing*, 22(12):1621–1645, 1997.