Stating and Manipulating Periodicity in the Polytope Model. Applications to Program Analysis and Optimization.

Expression et Manipulation de la Périodicité dans le Modèle Polyédrique. Applications à l'Analyse et à l'Optimisation de Programmes.

Benoît Meister
Équipe Image et Calcul Parallèle Scientifique
Laboratoire LSIIT, UMR 7005 ULP-CNRS
Université Louis Pasteur
Strasbourg, France
Thèse soutenue le 17 décembre 2004

Président du jury : Christine Eisenbeis, Directeur de Recherche à l'INRIA Futurs d'Orsay
Directeur de thèse :
Philippe Clauss, Professeur à l'Université Louis Pasteur de Strasbourg
Rapporteur interne :
Thomas Noël, Maître de conférences habilité à diriger des recherches à l'Université Louis Pasteur de Strasbourg
Rapporteurs externes :
Paul Feautrier, Professeur à l'École Nationale Supérieure de Lyon
Reinhard Wilhelm, Professeur, Universität des Saarlandes, Sarrebruck (Allemagne)
Examinateur :
Vincent Loechner, Maître de conférences à l'Université Louis Pasteur de Strasbourg

Pierrot (P. Dewaere): - "C'est pas mauvais cette petite potée."
Jean-Claude (G. Depardieu): - "Oh... pas de quoi écrire une thèse!"
Bertrand Blier, Les Valseuses (Going Places/Die Ausgebufften).

"La compilation, c'est bon."
Philippe Clauss.

Foreword/Avant-Propos

Ce manuscrit présente une grande partie des travaux de recherche que j'ai effectués pendant mes trois ans de thèse sous la direction de Philippe Clauss. C'est pour moi le premier ouvrage ayant nécessité un tel investissement, intellectuel et moral. Mais ces deux formes d'investissement complémentaires ne sont permises que sur des terrains favorables. Mes terrains à moi, ce qui m'enlève peut-être un peu de mérite, sont inépuisablement fertiles et généreux. C'est d'abord mon directeur de thèse, Philippe, et mon co-encadrant et ami, Vincent Loechner, qui ont fait germer la petite graine de chercheur que j'étais en licence et ont fait pousser mes premières feuilles (de papier). C'est ensuite tous ceux qui m'ont entouré de leur douce chaleur constante depuis le début, mes parents, mon frère, ma grand-mère, ma famille, mes amies, mes amis, ma petite amie, et les autres. C'est encore ceux qui m'ont illuminé de leur beauté d'esprit, les membres - permanents ou non, anciens ou non - de l'ICPS et leur inestimable ambiance de travail, et parmi eux Catherine Mongenet, qui rayonne d'énergie positive à la façon d'un petit soleil. C'est finalement les nombreux scientifiques que j'ai rencontrés lors de congrès et au conseil scientifique de l'Université Louis Pasteur, qui ont multiplié les branches scientifiques sur lesquelles portent mon intérêt. Parmi eux se distingue un petit groupe dont la tâche a été d'évaluer la maturité de ma production : mon jury de thèse. Sans oublier ceux que je n'ai que lus mais qui ont pris la peine d'être clairs, pédagogiques et honnêtes. Lecteur, si vous appréciez quelque partie de ce manuscrit, ayez une pensée en hommage à tous ceux sans qui vous ne pourriez goûter ce fruit.

Pour rendre également hommage aux choses abstraites, je dédie ce manuscrit de thèse à la scalabilité de la pensée, par laquelle une même chose peut paraître extrêmement importante et sérieusement insignifiante, et avec laquelle j'ai passé les quelques périodes hivernales de la thèse.

Contents

Chapter 1

Introduction

Moore's Law has been making admirable headway and presenting the computer industry with more and more gates on a chip. However, the same basic architecture of the 1960s is still being used to take advantage of these advances.

Microprocessor and DSP vendors are now hell-bent to use Moore's Law to ramp up their megahertz ratings or embedded multiple processors on the same die as a way to turbocharge their performance. But megahertz does not measure work done, and programmers have yet to be able to effectively use the underlying hardware and program more than two or three processors in parallel.

The traditional microprocessor and DSP are effectively hitting the proverbial wall. Large amounts of chip overhead and programming issues pose inherent limitations for conventional microprocessor and DSP architectures. Effort to improving microprocessor performance via architectural changes has waned because traditional chip designers are limited as to how to add gates efficiently in those architectures.

Compilers play a crucial role since they should be able to mask hardware specific functionalities as well as to do all the efficient program optimizations while relieving the programmers. This is of course ambitious. The ever growing complexity of hardware and application requirements poses huge difficulties for building good compilers. For example, real-time requirements imposes the necessity of precise execution time prediction and performance while hardware cost minimization involves efficient utilization of the small amount of resources.

Moreover in the past 10 years, the embedded industry has experienced an explosion.

Once relegated to massive infrastructure equipment, dedicated microprocessors are now pervasive, peppering everything from automobiles to personal digital media to toys with processing power. Application requirements for embedded systems induce more and more complexity, functionality, and sophistication.

All these trends create a need for compilers capable of generating high quality machine code. Such an objective can only be reached through smart program analysis and optimizations methods, capable of extracting accurate information related to the execution behavior on the target machine and capable of using efficiently this information to optimize the input source code.

This thesis wants to contribute by proposing an analysis framework for nested loops. It mainly focuses on the accuracy of the used geometrical model and particularly on the periodic aspect of loop analysis and transformations. Moreover it aims to expand the static analysis range through the development and the use of dedicated tools based on general mathematical theories. This work is representative of the approach consisting in expanding as far as possible the range of static analysis rather than using less general and less accurate dynamic approaches motivated by the "too high" complexity of the underlying mathematical models.

1.1 Specific background

For many years, compiler writers have focused on parameterized loop nests, mainly because of their importance in scientific and multimedia programs. The *polytope model* [4] 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 1.1. 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

$$P = \begin{cases} 1 \leq i \leq n \\ i+1 \leq j \leq n \\ i+1 \leq k \leq n \end{cases}$$

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 [85, 31, 52, 86, 32, 87, 28]), loop nests are usually transformed by applying an affine integer transformation to the $\mathbb{Z}-$polytope representing the loop nest [74]. They may also be split into sub-polyhedra as for example in [37]. 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 $P \cap L$. To have a better insight of the problem, let us consider the polytope presented in example 1.2.

Example 1.2. The following polytope is represented on figure 1.1.

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

□

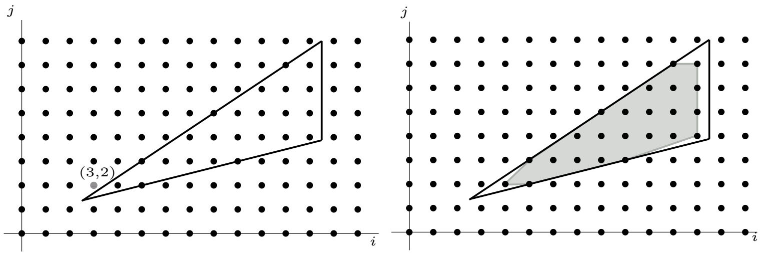

Figure 1.1: P_2, the wrong lexicographic minimum, and $int(P_2)$

Its corresponding loop nest in the C language, generated by using the Fourier-Motzkin algorithm while choosing i as the outer loop index, 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 1.1 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 $int(P_2)$ of P_2 (in gray on figure 1.1), as its vertices are integer while it includes all the integer points (iterations) of P_2. Here, $int(P_2)$ corresponds to the following polyhedron:

$$int(P_2) = \begin{cases} 2i - 3j - 1 \geq 0 \\ -j + 7 \geq 0 \\ i - j - 2 \geq 0 \\ -i + 4j - 3 \geq 0 \\ j - 2 \geq 0 \\ -i + 3j \geq 0 \\ -i + 12 \geq 0, \end{cases}$$

whose Fourier-Motzkin projection gives the following loop nest:

```
for (i=4; i<= 12; i++)
  for (j=max(2, ceil(i/3), ceil((i-3)/4)) ; j<=min(7, floor((2i-1)/3), i-2) ; j++)
    /* statements */
```

The lexicographic minimum, $\{i = 4; j = 2\}$ is directly given by the lower loop bounds.

This thesis focuses on manipulating integer points that belong to a parametric rational polyhedron P. In chapter 2, we first devise an algorithm for the convex hull of all these points, i.e., the *integer hull* of P. Then, we consider the problem of finding particular integer points. The points we look for have to be maximum w.r.t. a given *hierarchically linear order*. This order encompasses the classical linear order as well as the lexicographic order (which represents the execution order of the iterations in a loop nest). We propose an algorithm for computing such a maximum in chapter 3. This result is exploited to determine the integer points in the projection of a $\mathbb{Z}$-polyhedron, and to count these points. Then, we devote chapter 4 on the study of Ehrhart's conjecture, which defines a link between the faces of P and the coefficients of the *Ehrhart polynomial* of P, which gives the number of integer points in P as a function of P's parameters. Approximation methods for Ehrhart polynomials are then proposed, as well as an optimization for an existing algorithm for computing Ehrhart polynomials, and a new method for computing Ehrhart polynomials in some specific cases. We describe in chapter 5 what we have implemented among the concepts and algorithms we present across the thesis. An overview of the work is finally given in chapter 6, as well as future working directions.

But first, some mathematical preliminaries are given in the next section.

1.2 Preliminaries

1.2.1 Normal forms

Two integer matrix decompositions are commonly used in the polytope model: the Hermite and Smith normal forms. Before presenting these decompositions and their properties, we must recall the terms of *unimodular matrix* and *full-rank* matrix.

In the frame of this thesis, a unimodular matrix denotes a square integer matrix whose determinant is ± 1. The (row- or column-) vectors of a n-dimensional unimodular matrix span $\mathbb{Z}^n$. A unimodular matrix is invertible, and its inverse is also unimodular.

A matrix is of full row-rank if the number of its non-zero rows cannot be reduced by row-elimination. Similarly, a matrix is of full column-rank if the number of its non-zero columns cannot be reduced by column-elimination.

There exist two Hermite normal forms: the left Hermite normal form of a full row-rank matrix A corresponds to the following decomposition:

$$A = [H_A \ 0]U_A.$$

The properties that will be used in this thesis are:

- H_A is an integer lower triangular matrix of nonnegative elements, and it is invertible,
- U_A is an integer unimodular matrix,
- the column-vectors of A and H_A span the same integer lattice.

The right Hermite normal form is similar to the left form. For a full column-rank matrix A, it is:

$$A = U_A \begin{bmatrix} H_A \\ 0 \end{bmatrix}.$$

U_A is unimodular and H_A is an invertible integer upper triangular matrix of nonnegative elements. Also, the row-vectors of A and H_A span the same integer lattice.

The Smith normal form of an integer matrix A is the following decomposition:

$$A = U_A \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} V_A,$$

where U_A and V_A are unimodular and S is an integer diagonal matrix of positive diagonal elements.

More details and properties of these normal forms are given in [76].

1.2.2 A regularly occurring problem

At several occasions across this thesis, we will need to know under which condition there exists an integer solution I to the set of m parametric equalities:

$$AI + BN + C = 0 \tag{1.1}$$

where $N \in \mathbb{Z}^p$ are the parameters of the problem, and A, B and C are respectively $(m \times n), (m \times p)$ and $(m \times 1)$ integer matrices. There is an integer solution to equation (1.1) if the m-dimensional integer point $(BN+C)$ belongs to the lattice of integer points spanned by the column-vectors of A. In other words, there must be an integer linear combination (given by I) of the column-vectors of A that is equal to $BN + C$.

Let $A = [H_A\ 0]U_A$ be the Hermite normal form of A. As the column-vectors of H_A and A span the same lattice, the condition for equation (1.1) to have a solution is that there exists $X \in \mathbb{Z}^m$ such that: $H_A X + (BN + C) = 0$. As H_A is invertible, this can be written $H_A^{-1}(BN + C) = -X$, or similarly

$$H_A^{-1}(BN + C) \in \mathbb{Z}^m.$$

1.2.3 Validity domains

We are going to work on a parametric polyhedron P. In [60, 55], Loechner and Wilde have pointed out that the shape of P depends on the value of the parameters. They have shown that the parameters space can be partitioned into adjacent rational polyhedra, called *validity domains*, in which P has a given shape. An algorithm for computing these validity domains is detailed in [55]. A given shape determines:

- a given affine expression for the vertices of P,
- a given set of constraints which are non-redundant in the corresponding validity domain.

Chapter 2

A method for computing the integer hull of a rational parametric polyhedron

In the polytope model, iterations of a loop nest are modeled by integer points of a rational parametric polyhedron P. The variables of the program that are unknown at compile-time are treated as parameters. Besides, compilation parameters can be taken into account, as for instance an assumed execution context. This might include memory, cache, TLB and register file size, cache associativity, maximum execution time or energy consumption and availability of other resources. For a more direct quantitative and qualitative analysis of nested loops, we want to characterize the whole set of integer points of P by their convex hull, which is called the *integer hull* of P, in order to make somehow clearer the problems related to integer points in a rational parametric polyhedron. Another motivation is that, to our knowledge, the rare existing algorithms for computing the integer hull of a polyhedron are too restrictive. In particular, they are not parametric. It is known that the computational complexity of all the integer hull algorithms (including ours) is exponential in function of the number of variables and parameters. Then, it has become usual to look at their complexity *for a fixed dimension*, i.e., when the number of variables and parameters are fixed. The first algorithm is due to Schrijver [76], chap. 23, who uses Gomory's cutting planes. In [29], Feautrier extends the cutting planes to the case of parametric polyhedra in the context of integer linear programming. Extending Schrijver's integer hull algorithm to the parametric case could then be envisaged. In [7], Bockmayr and Eisenbrand recall that the complexity of this algorithm is exponential for a fixed dimension. They present a polynomial-time algorithm (for a fixed dimension) for finding the *elementary closure* of P, which represents all the possible cutting planes for P. The extension of this algorithm to the parametric case could help to find a polynomial-time algorithm for computing the integer hull of P for a fixed dimension. However, this extension is far from being obvious. Besides, Harvey [41] gives a fast algorithm for computing the integer hull of a 2-dimensional polytope, based on unimodular transformations. He then uses this algorithm in the context of logic programming expressed with constraints, in the particular case where each constraint depends on two variables. The work of Lasserre [50], which is concurrent with the one described in this chapter, showed that a $\mathbb{Z}$-polyhedron defined as $\{Ax + b = 0, x \in \mathbb{N}^n\}$ can be transformed into a polyhedron defined as $\Omega = \{Mq + r = 0, q \in \mathbb{R}^+\}$, where M is totally unimodular and where r is integer, which implies that Ω has only integer vertices. A linear relationship $x = Eq$ exist between x and q. The constraints of the integer hull are then derived from a convex cone built with M and E. The extension to the parametric case is unlikely to be easy.

In this chapter, we present a new approach for computing the integer hull P' of a rational parameterized polyhedron 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.

For simplicity, we first assume the following restrictions on the considered parametric polyhedron:

- it is bounded: it is a parametric *polytope*,
- the values for which it is not empty (its definition domain) belongs to only one Loechner-Wilde validity domain (see section 1.2),
- it is full-dimensional: it is defined only by a set of inequalities, and there is no implicit equalities for the values of parameters that belong to the definition domain of P: a subset of the inequalities defining P can never be equivalent to an equality.

An intuitive presentation of the method is given in next section. Then, section 2.2 introduces the mathematical objects used in section 2.3 to turn the problem into a geo-

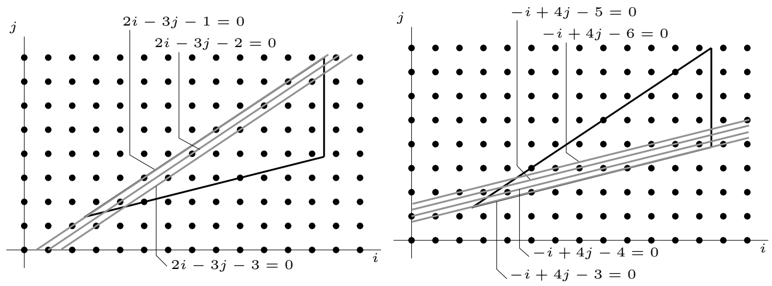

Figure 2.1: Integer minima and maxima for j for a given value of i

metric form. An algorithm in devised in section 2.4. The assumed restrictions on P are discussed in section 2.5.

2.1 Basic Idea

Consider the polytope P_2 represented with its integer hull on figure 1.1. The maximum integer value j_{max} of j for a given i in P_2 is determined by one of its defining inequalities:

$$2i - 3j - 1 \geq 0 \Leftrightarrow j \leq \frac{2i-1}{3}$$

As we can see on figure 2.1, when $i\ mod\ 3 = 0$, j_{max} is solution to the equation $2i-3j-3 = 0$. It is solution to $2i-3j-1=0$ when $i\ mod\ 3 = 2$, and solution to $2i-3j-2=0$ when $i\ mod\ 3 = 1$. Similarly, the minimal integer value j_{min} is determined by the inequality $\{-i+4j-3 \geq 0\}$. It is solution to

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

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 reader can figure out by comparing figures 2.2 and 1.1 that the convex hull of all these extremal points (drawn in gray) is the integer hull of P_2.

To generalize this idea to n dimensions, consider that any integer point in a n-dimensional space

$$X = (x_1, x_2, \ldots, x_{k-1}, x_k, x_{k+1}, \ldots, x_n)$$

of a polyhedron P belongs to a line segment $d_k(X)$, which is the intersection of a line and P, defined by

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

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 $i_k \leq p'_{max}(I, N)$, where $p'_{min}(I, N) = -\frac{p_{min}(I,N)}{a_k} + i_k$ and $p'_{max}(I, N) = -\frac{p_{max}(I,N)}{a_k} + i_k$. 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

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

and

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

It follows that:

- the convex hull P' of all the existing extremal points, for any $X \in P$, $x_{min}(X, N)$ and $x_{max}(X, N)$ contains all the integer points of P,
- $P' \subset P$,
- P' has integer vertices, which are some of the extremal points,

which implies that P' is the integer hull of P.

Example 2.1. The existing extremal points $x_{min}(i_1)$ and $x_{max}(i_1)$ are represented on figure 2.2, as well as the different $d_2(i_1, i_2)$. Observe that some of the extremal points are the vertices of the integer hull of P_2, and that the convex hull of all these points is the integer hull of P_2 (represented on figure 1.1). □

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

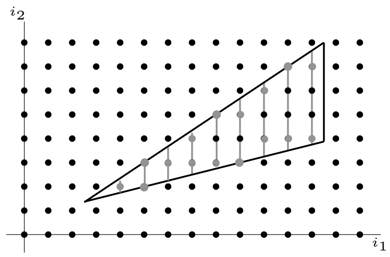

Figure 2.2: The extremal integer points of P_2 and the different $d_2(i_1, i_2)$

2.2 Periodic Polyhedra

Consider the k^{th} coordinate $x_{k,max}(X, N)$ of $x_{max}(X, N)$ determined by the inequality

$$p_{max}(X, N) = \sum_{t=1}^{n} a_t x_t + \sum_{u=1}^{p} b_u n_u + c \geq 0 \ :$$

$$x_{k,max}(X, N) = \lfloor p'_{max}(X, N) \rfloor.$$

with $p'_{max}(X, N) = \frac{p_{max}(X,N)}{-a_k} + x_k$.

Example 2.2. Consider the inequality $p_{max}(i, j, m) = \{-3i + 2j - m + 5 \geq 0\}$, which defines an upper bound on i:

$$i \leq \frac{2j - m + 5}{3}$$

Its maximal integer value is given by

$$i_{max}(i, j, m) = \lfloor \frac{2j - m + 5}{3} \rfloor = \lfloor p'_{max}(i, j, m) \rfloor$$

□

Let α be an integer and β a positive integer. By defining the *mod* operator as the remainder of a division by an integer:

$$\forall \alpha \in \mathbb{Q}, \beta \in \mathbb{N}, 0 \leq \alpha' < \beta \in \mathbb{Q}^+ : \alpha \ mod \ \beta = \alpha' \Leftrightarrow \alpha = k.\beta + \alpha', k \in \mathbb{Z}, 0 \leq \alpha' \leq \beta$$

the following formulae come:

$$\lfloor \frac{\alpha}{\beta} \rfloor = \frac{\alpha}{\beta} - \frac{\alpha \ mod \ \beta}{\beta},$$

$$\lceil \frac{\alpha}{\beta} \rceil = \frac{\alpha}{\beta} + \frac{(-\alpha)\ mod\ \beta}{\beta}.$$

Hence, we can write: $x_{k,max} = \lfloor \frac{-a_k.p'_{max}(X,N)}{-a_k} \rfloor$, and since $a_k < 0$:

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

$$\Leftrightarrow -a_k x_{k,max} = -a_k.p'_{max}(X,N) - (-a_k.p'_{max}(X,N))\ mod\ (-a_k).$$

As $-a_k.p'_{max}(X,N) = p_{max}(X,N) - a_k x_k$, it reduces to:

$$p_{max}(X,N) - (p_{max}(X,N) - a_k x_k)\ mod\ (-a_k) = a_k(x_k - x_{k,max}).$$

This gives the necessary and sufficient condition for x_k to reach its maximal integer value w.r.t. the inequality $p_{max}(X,N) \geq 0$:

$$x_k = x_{k,max} \Leftrightarrow p_{max}(X,N) - p_{max}(X,N)\ mod\ (-a_k) = 0 \tag{2.1}$$

Similarly, we have:

$$x_k = x_{k,min} \Leftrightarrow p_{min}(X,N) - p_{min}(X,N)\ mod\ (a_k) = 0 \tag{2.2}$$

Theorem 2.1 follows equations (2.1) and (2.2):

Theorem 2.1. *Let $p(I,N) \geq 0$ be an inequality with integer coefficients:*

$$p(I,N) = \sum_{t=1}^{n} a_t i_t + \sum_{u=1}^{p} b_u n_u + c \geq 0.$$

Let $X = (x_1, x_2, \ldots, x_n) \in \mathbb{Z}^n$. The extremal possible integer value of x_k satisfying $p(X,N) \geq 0$ for integer values of $(x_1, x_2, \cdots, x_{k-1}, x_{k+1}, \cdots, x_n)$ is solution to

$$\mathcal{I}(p(X,N),k) = 0, \tag{2.3}$$

where $\mathcal{I}(p(X,N),k) = p(X,N) - p(X,N)\ mod\ (|a_k|)$.

In this chapter, $\mathcal{I}(p(X,N),k)$ will be called the *integer bound* of x_k w.r.t. $p(X,N)$. So the set of extremal points $x_{min}(X,N)$ and $x_{max}(X,N)$ are integer solutions to (2.3) for some constraint $p(X,N) \geq 0$ of P. It is known in theory of numbers that an affine function of an integer variable is periodic modulo $m \in \mathbb{N}$ (see for instance [78]), which explains the periodic character (*periodicity*) of the extremal points. We choose to handle periodicity by using a tailored class of mathematical objects, called *periodics*. We believe that they are more appropriate than modulo congruences to understand the geometric approach we propose. Periodics are presented in [66] as a generalization of *periodic numbers*, initially introduced by Ehrhart [26] and extended by Clauss [16] as coefficients for Ehrhart polynomials. A periodic over a monoïd K is defined as a periodic function of integer variables whose values are in K. In this chapter, only three instances of K are considered: rational numbers, polyhedra and polynomials.

Let $f(I), I \in \mathbb{Z}^n$ be an affine function with integer coefficients and let $m \in \mathbb{Z}$. $f(I)\ mod\ m$ is an integer *periodic number*. A periodic number of period $S = \begin{pmatrix} s_k \end{pmatrix} \in \mathbb{N}^n$ is a rational-valued periodic function of I. 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 2.3. A 2-dimensional periodic number of period $S = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$ depending 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 $N \bmod 2 = 1$, $M \bmod 3 = 2$

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

Consider $A(i) = i\ mod\ 3, i \in \mathbb{Z}$. 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 $m \in \mathbb{N}$, $f(I)\ mod\ m$ is an integer periodic number of period $S = (s_k)$ with $s_k = \frac{m}{gcd(m,|a_k|)}$.

Example 2.4. With $f(I) = 3i + 4j$, and $m = 6$, it gives:

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

which is an integer periodic number of period $S = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$. □

The arithmetic operators on rational numbers can be translated in operators on periodic numbers. Let $\otimes$ be an arithmetic operator on rational numbers. Let A and B be n-dimensional (without loss of generality) periodic numbers of respective periods $S^A \in \mathbb{Z}^n$ and $S^B \in \mathbb{Z}^n$. The periodic number C resulting from the operation $\otimes$ extended to periodic numbers:

$$C = A \otimes B$$

is defined for any $I \in \mathbb{Z}^n$ by:

$$C(I) = A(I\ mod\ S^A) \otimes B(I\ mod\ S^B).$$

Its period is $S = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} \in \mathbb{Z}^n$ such that $s_k = lcm(s_k^A, s_k^B)\ \forall k \in [1, n]$. This extension is quite straightforward from rational numbers to periodic numbers. However, we will see in next section that an operator on a monoid K can not always be straightforwardly translated in an operator on periodics over K.

Using periodic numbers, we can give an explicit form of the equalities yielding the extremal points.

Example 2.5. According to theorem 2.1, coordinate j of the extremal points of P_2 is given by one of the two equalities:

$$-i + 4j - 3 - (-i + 4j - 3)\ mod\ 4 = -i + 4j - 3 - \begin{bmatrix} 1 & 0 & 3 & 2 \end{bmatrix}_i = 0$$

$$2i - 3j - 1 - (2i - 3j - 1)\ mod\ 2 = 2i - 3j - 1 - \begin{bmatrix} 1 & 0 \end{bmatrix}_i = 0,$$

which is concordant with our preliminary observations. □

The two forms of a periodic number, an affine function modulo an integer and an array of integers, are equivalent. However, to distinguish them, the first form will be called *symbolic form*[1] and the second form will be called *explicit form* (as its values are given explicilty).

The 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*. It can be defined as a periodic function whose values are polyhedra. But in order to offer the right geometric intuition, we use the following equivalent geometric definition (stated in [66]):

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

- *$q = s_1 \times s_2 \times \ldots \times s_n$ polyhedra M_I, indexed by:*

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

- *their respective definition lattice $\mathcal{L}_I$: the integer lattice defined by:*

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

$\bigcup_I \mathcal{L}_I = \mathbb{Z}^n$: any element $X \in \mathbb{Z}^n$ is mapped to a unique M_I by the relation [2] *$I = X \bmod S$.* □

Notice that each M_I is then the intersection between an integer lattice and a polyhedron (i.e. a $\mathbb{Z}-$polyhedron). As well as periodic numbers, periodic polyhedra can be represented as a n-dimensional array of polyhedra.

Example 2.6. A 2-dimensional periodic polyhedron (made of one inequality) of $(i, j, k) \in \mathbb{Z}^3$ which is periodic along on $(i, j) \in \mathbb{Z}^2$, whose value is :

[1] we will see later that, in our frame, the general symbolic form of a periodic number is actually more complicated than just an affine function modulo an integer

[2] The reader used to the theory of numbers can notice that the $\mathcal{L}_i$'s correspond to the direct sum of the n residue classes modulo $s_k, k \in [1..n]$, which are known to partition $\mathbb{Z}^n$

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

can be represented as:

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

, 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 *factored form*, as the constant form of an affine function of (i, j, k) is extracted. □

Using periodic polyhedra, we show in next section new geometric tools that will allow to compute the integer hull of a parametric rational polyhedron.

2.3 Pseudo-facets

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

Definition 2. *Let $p_q(I, N) \geq 0$ be an inequality of P. The q^{th} facet f_q of a polyhedron P can be defined by:*

$$\{I \in P \mid p_q(I, N) = 0\}.$$

Similarly, the q^{th} pseudo-facet *$f'_{q,k}$ of P w.r.t. i_k is defined by:*

$$\{I \in P \mid \mathcal{I}(p_q(I, N), k) = 0\}.$$

To get further into the comparison between a facet and a pseudo-facet, notice that the facet f_q can be decomposed as the intersection of two polyhedra:

- a hyperplane, defined by $p_q(I, N) = 0$, which by the way defines the affine hull of f_q,
- and a *projected* polyhedron, which is obtained by eliminating a variable, say i_k, in P by using $p_q(I, N) = 0$. The result is called *projected* as it corresponds to the projection along i_k of f_q.

Similarly, a pseudo-facet $f'_{q,k}(P)$ can be decomposed as the intersection of two periodic polyhedra:

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

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

Example 2.7. Consider the parameterized polyhedron P_3 defined by:

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

Figure 2.7 shows the facet of P defined by :

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

which can be decomposed into a (supporting) hyperplane and a $n-1$-dimensional polyhedron:

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

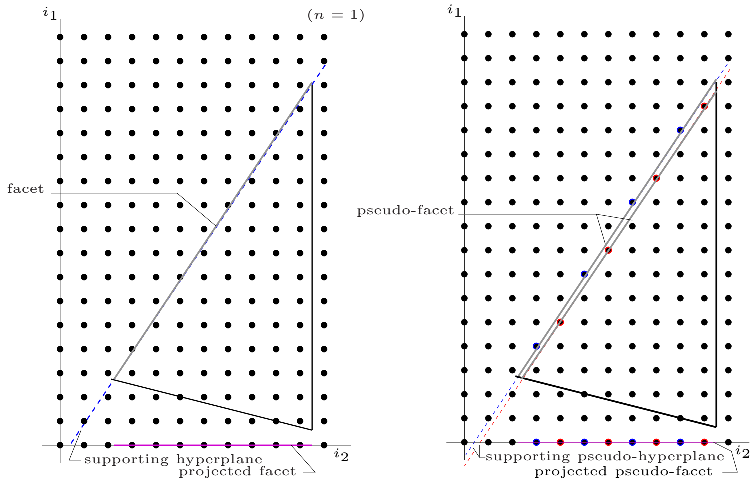

Figure 2.3: A facet of P_3 and its corresponding pseudo-facet w.r.t. i_1.

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

$$\begin{cases} \mathcal{I}(-2i_1 + 3i_2 - n, 1) = -2i_1 + 3i_2 - n - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}_{i_2,n} = 0 \\ -2i_2 + 21 \geq 0 \\ 4i_1 + j_2 - 13 \geq 0 \end{cases},$$

which can be decomposed into a periodic hyperplane and a periodic $n-1$-dimensional polyhedron, the projected pseudo-facet:

$$= \begin{cases} -2i_1 + 3i_2 - n - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}_{i_2,n} = 0 \\ \begin{cases} -2i_2 + 21 \geq 0 \\ 7i_2 - 2n - \begin{bmatrix} 13 & 15 \\ 15 & 13 \end{bmatrix}_{i_2,n} \geq 0 \end{cases} \end{cases}$$

□

We can state the basic idea for computing the integer hull of a rational polytope using our brand new geometric object:

Theorem 2.2. *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. According to theorem 2.1, the extremal points, whose convex hull is the integer hull of P, belong to pseudo-facets of P. Moreover, each of the extremal points belongs to the integer hull of a pseudo-facet of P w.r.t. i_k. The convex hull of the integer hulls of all the pseudo-facets is then the integer hull of the extremal points, that is to say the integer hull of P. □

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. Following the recursive theorem 2.2, the pseudo-facets of the pseudo-facets of P have then to be computed. But the pseudo-facets of P are periodic polyhedron, for which pseudo-facets are not defined yet. A naïve manner to compute the pseudo-facets of a periodic polyhedron P' would be to consider each possible definition of P' (which is a polyhedron) given by its explicit form, and to compute their individual pseudo-facets. In [66], we show that operators on periodics over a monoid K cannot in general be derived straightforwardly from the operators on K. In next section, this issue is tackled for periodic polyhedra: some operators on polyhedra cannot be translated naively into a semantically equivalent operator on periodic polyhedra.

2.3.1 Pseudo-facets of periodic polyhedra

As seen in definition 1, a periodic polyhedron P has the same definition P_{J_0} for each point belonging to the integer lattice

$$SJ + J_0 = \begin{pmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & s_{n+p} \end{pmatrix} J + J_0, J \in \mathbb{Z}^{n+p}.$$

It is indexed by J_0, and its determinent, $q = s_1 \times s_2 \times \cdots \times s_{n+p}$, is greater than 1 if P is periodic. Computing directly the integer bound of a variable w.r.t. some constraint for each polyhedron P_{J_0} may lead to points that do not belong to this lattice.

Example 2.8. Consider the following periodic polyhedron:

$$\left\{ -2i + \begin{bmatrix} 19 & 20 \end{bmatrix}_i \geq 0 \right.$$

Let us compute an upper bound of i for each of the possible definitions the polyhedron:

- when $i \ mod \ 2 = 0$:

$$-2i_{max} + 19 - (-2i_{max} + 19) \ mod \ 2 = 0 \Rightarrow i_{max} = 9$$

- when $i \ mod \ 2 = 1$:

$$-2i_{max} + 20 - (-2i_{max} + 20) \ mod \ 2 = 0 \Rightarrow i_{max} = 10$$

The computed integer bounds are contradictory with the conditions on i: i_{max} is odd when it is supposed to be even, and conversely.

□

Hence, we must consider the integer part *on the lattice* where the constraint is defined. This can be enforced by the following variable substitution: $\begin{pmatrix} I \\ N \end{pmatrix} = SJ + J_0$, where $J_0 = (j_k)$, $j_k = i_k \ mod \ s_k$ for $k \in [1..n]$ and $j_k = n_k \ mod \ s_k$ for $k \in [n+1..n+p]$. Considering J_0 as parameters, the supporting lattice for the variables is then $\mathbb{Z}^n$: an integer bound operation can only result in some point of this new lattice.

Then, taking the integer part of i_k w.r.t. the constraint $\sum_l a_l i_l + c$ of P leads to compute:

$$\mathcal{I}(\sum_l a_l i_l + \sum_{m=n+1}^{n+p} b_m n_m + c, k) = \mathcal{I}(\sum_l a_l (s_l j_l + j_{0,l}) + \sum_{m=n+1}^{n+p} b_m (s_m j_m + j_{0,m}) + c, k)$$

$$= \sum_l a_l (s_l j_l + j_{0,l}) + \sum_{m=n+1}^{n+p} b_m (s_m j_m + j_{0,m}) + c$$

$$-((\sum_l a_l (s_l j_l + j_{0,l}) + \sum_{m=n+1}^{n+p} b_m (s_m j_m + j_{0,m}) + c) \ mod \ |a_k s_k|)$$

$$= \sum_l a_l i_l + \sum_{m=n+1}^{n+p} b_m n_m + c - ((\sum_l a_l i_l + \sum_{m=n+1}^{n+p} b_m n_m + c) \ mod \ |a_k s_k|)$$

Hence, the definition of a pseudo-facet of a periodic polyhedron is the same as for a polyhedron, except that the *integer bound* operator, $\mathcal{I}$, is a bit different, as it takes the existing period of i_k into account. Notice that this new definition of $\mathcal{I}$ is just a generalization to periodic polyhedra: a polyhedron is nothing else than a periodic polyhedron with $s_k = 1, k \in [1..n]$.

Example 2.9. The following periodic polyhedron with variables $I = \begin{pmatrix} i_1 \\ i_2 \end{pmatrix}$ and parameters $N = \begin{pmatrix} n \end{pmatrix}$:

$$P_1(i_1, j_1, n) = \begin{cases} -2i_1 + 3i_2 - n - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}_{i_2,n} = 0 \\ -2i_2 + 21 \geq 0 \end{cases}$$

has a period $S = \begin{pmatrix} s_{i_1} = 1 \\ s_{i_2} = 2 \\ s_n = 2 \end{pmatrix}$. Its pseudo-facet obtained from the constraint $-2i_2 + 21 \geq 0$ w.r.t. i_2 is then:

$$f'_{2,2} = \begin{cases} -2i_1 + 3i_2 - n - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}_{i_2,n} = 0 \\ -2i_2 + 21 - (-2i_2 + 21 \; mod \; |-2s_{i_2}|) = 0 \end{cases}$$

which can be written:

$$\begin{cases} -2i_1 + 3i_2 - n - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}_{i_2,n} = 0 \\ -2i_2 + 21 - \begin{bmatrix} 1 & 3 \end{bmatrix}_{i_2} = 0 \end{cases}$$

□

The variable substitution technique allows to avoid the issues due to periodicity. However, one can notice that periodicity has no impact on some operators, for instance the convex hull operator for a periodic polyhedron.

Now that the meaning of theorem 2.2 is clear, we can devise an algorithm for computing the integer hull of a parametric polyhedron P.

2.4 Integer hull

Theorem 2.2 states a relation between the integer hull $int(P)$ of a polyhedron P and the integer hull of its pseudo-facets w.r.t. variable i_k:

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

where $conv(X)$ denotes the convex hull of a periodic polyhedron X.

Let us see what happens when getting into the recursion: 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:

$$int(f'_{q,k}) = conv\left(\bigcup_{q'} int(f''_{q',k'}(f'_{q,k}))\right) \tag{2.4}$$

According to theorem 2.1, 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 and parameters 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 if the parameters are integer (which is assumed). 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 $int(P)$ are obtained by scanning this tree from its root P to its 0-dimensional pseudo-facets, called the *pseudo-vertices* of P. 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 = 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''
}
```

Example 2.10. Let us compute the pseudo-vertices of

$$P_3 = \begin{cases} -2i_1 + 3i_2 - n \geq 0 & (a) \\ -2i_2 + 21 \geq 0 & (b) \\ 4i_1 + i_2 - 13 \geq 0 & (c) \end{cases}$$

where i_1, i_2 are the (respectively first and second) variables and n is the parameter. Each inequality has been marked with a letter. The pseudo-facets we are going to compute are named according to the inequalities that are transformed into equalities. The order of indices in which we choose to compute the pseudo-facets is i_1 and then i_2.

Pseudo-facet (a) w.r.t. i_1 has been computed in example 2.7:

$$(a) = \begin{cases} -2i_1 + 3i_2 - n - (2i_1 + 3i_2 - n)\ mod\ 2 = 0 \\ \begin{cases} -2i_2 + 21 \geq 0 \\ 7i_2 - 2n - 13 - 2[(2i_1 + 3i_2 - n)\ mod\ 2] \geq 0 \end{cases} \end{cases}$$

The period of (a) is $S = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}$. Two pseudo-facets of (a) w.r.t. (i_2) can be derived:

$$(ab) = \begin{cases} -2i_1 + 3i_2 - n - (3i_2 - n)\ mod\ 2 = 0 \\ -2i_2 + 21 - (-2i_2 + 21)\ mod\ (2 \times 2) = 0 \\ \begin{cases} -4n + 121 - 7[(-2i_2 + 21)\ mod\ 4] - 4[(3i_2 - n)\ mod\ 2] \geq 0 \end{cases} \end{cases}$$

$$= \begin{cases} -2i_1 + 3i_2 - n - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}_{i_2,n} = 0 \\ -2i_2 + 21 - \begin{bmatrix} 1 & 3 \end{bmatrix}_{i_2} = 0 \\ \begin{cases} -4n + 121 - \begin{bmatrix} 7 & 11 \\ 25 & 21 \end{bmatrix}_{i_2,n} \geq 0 \end{cases} \end{cases}$$

$$(ac) = \begin{cases} -2i_1 + 3i_2 - n - (3i_2 - n) \ mod \ 2 = 0 \\ 7i_2 - 2n - 13 - 2[(3i_2 - n) \ mod \ 2] \cdots \\ \cdots - [7i_2 - 2n - 13 - 2[(3i_2 - n) \ mod \ 2] \ mod \ (7 \times 2)] = 0 \\ \begin{cases} -4n + 121 - 4[(3i_2 - n) \ mod \ 2] \cdots \\ \cdots - 2[7i_2 - 2n - 13 - 2[(3i_2 - n) \ mod \ 2] \ mod \ 14] \geq 0 \end{cases} \end{cases}$$

$$= \begin{cases} -2i_1 + 3i_2 - n - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}_{i_2,n} = 0 \\ 7i_2 - 2n - 13 - \begin{bmatrix} 1 & 13 & 11 & 9 & 7 & 5 & 3 & 15 & 13 & 11 & 9 & 7 & 5 & 3 \\ 8 & 6 & 4 & 2 & 14 & 12 & 10 & 8 & 6 & 4 & 2 & 0 & 12 & 10 \end{bmatrix}_{i_2,n} = 0 \\ \begin{cases} -4n + 121 - \begin{bmatrix} 2 & 26 & 22 & 18 & 14 & 10 & 6 & 30 & 26 & 22 & 18 & 14 & 10 & 6 \\ 16 & 12 & 8 & 4 & 28 & 24 & 20 & 16 & 12 & 8 & 4 & 0 & 24 & 20 \end{bmatrix}_{i_2,n} \geq 0 \end{cases} \end{cases}$$

For pseudo-facet (b) w.r.t. i_1, consider the equality

$$\mathcal{I}(-2i_2 + 21 \geq 0, 1) = -2i_2 + 21 - (-2i_2 + 21) \ mod \ 0,$$

which does not give an extremal value of i_1, as the inequality of P_3 marked with (b) is independent from i_1. Thus, there is not pseudo-facet (b) w.r.t. i_1. Finally, let us compute pseudo-facet (c) w.r.t. i_1:

$$(c) = \begin{cases} 4i_1 + i_2 - 13 - (i_2 - 13) \ mod \ 4 = 0 \\ \begin{cases} 7i_2 - 2n - 13 - (i_2 - 13) \ mod \ 4 \geq 0 \\ -2i_2 + 21 \geq 0 \end{cases} \end{cases},$$

of period $\begin{pmatrix} 1 \\ 4 \\ 1 \end{pmatrix}$. Pseudo-vertices (ca) and (cb) can be derived:

$$(ca) = \begin{cases} 4i_1 + i_2 - 13 - (i_2 - 13) \ mod \ 4 = 0 \\ 7i_2 - 2n - 13 - (i_2 - 13) \ mod \ 4 \cdots \\ \cdots - [7i_2 + 2n - 13 - (i_2 - 13) \ mod \ 4] \ mod \ (7 \times 4) = 0 \\ \begin{cases} 4n + 121 - 2[(i_2 - 13) \ mod \ 4] \cdots \\ \cdots - 2[[7i_2 + 2n - 13 - (i_2 - 13) \ mod \ 4] \ mod \ 28] \geq 0 \end{cases} \end{cases}$$

$$= \begin{cases} 4i_1 + i_2 - 13 - \begin{bmatrix} 3 & 0 & 1 & 2 \end{bmatrix}_{i_2} = 0 \\ 7i_2 - 2n - 13 - \begin{bmatrix} 15 & 13 & 11 & 9 & 7 & 5 & 3 & 29 & 27 & 25 & 23 & 21 & 19 & 17 \\ 22 & 20 & 18 & 16 & 14 & 12 & 10 & 8 & 6 & 4 & 2 & 0 & 26 & 24 \\ 1 & 27 & 25 & 23 & 21 & 19 & 17 & 15 & 13 & 11 & 9 & 7 & 5 & 3 \\ 8 & 6 & 4 & 2 & 28 & 26 & 24 & 22 & 20 & 18 & 16 & 14 & 12 & 10 \end{bmatrix}_{i_2,n} = 0 \\ \begin{cases} 4n + 121 - \begin{bmatrix} 30 & 26 & 22 & 18 & 14 & 10 & 6 & 58 & 54 & 50 & 46 & 42 & 38 & 34 \\ 44 & 40 & 36 & 32 & 28 & 24 & 20 & 16 & 12 & 8 & 4 & 0 & 52 & 48 \\ 2 & 54 & 50 & 46 & 42 & 38 & 34 & 30 & 26 & 22 & 18 & 14 & 10 & 6 \\ 16 & 12 & 8 & 4 & 56 & 52 & 48 & 44 & 40 & 36 & 32 & 28 & 24 & 20 \end{bmatrix}_{i_2,n} \geq 0 \end{cases} \end{cases}$$

$$(cb) = \begin{cases} 4i_1 + i_2 - 13 - (i_2 - 13)\ mod\ 4 = 0 \\ -2i_2 + 21 - (-2i_2 + 21)\ mod\ (2 \times 4) = 0 \\ \{4n + 121 - 7(-2i_2 + 21)\ mod\ 8 - 2[(i_2 - 13)\ mod\ 4] \geq 0 \end{cases}.$$

$$= \begin{cases} 4i_1 + i_2 - 13 - \begin{bmatrix} 3 & 0 & 1 & 2 \end{bmatrix}_{i_2} = 0 \\ -2i_2 + 21 - \begin{bmatrix} 5 & 3 & 1 & 7 \end{bmatrix}_{i_2} = 0 \\ \{-4n + 121 - \begin{bmatrix} 41 & 21 & 9 & 53 \end{bmatrix}_{i_2} \geq 0 \end{cases}.$$

As this description of a pseudo-vertex can be represented as a (upper) triangular matrix (without the domain of parameters), it is called the *triangular form* of a pseudo-vertex. The coordinates of the pseudo-vertices are derived from their triangular form by row-elimination. For this example, they are written in their explicit form on figure 2.4.

□

Notice that the periodic numbers of the pseudo-vertices depend on the variables. Variables can also be instantiated in the periodic numbers, giving a set of extremal points which are periodic along the parameters only. The number of extremal points generated by instantiating a variable i_k equals the period of the pseudo-vertex along this variable.

Example 2.11. Pseudo-vertex (ab) still depends on variable i_2:

$$\begin{cases} i_1 = (-2n + 63 - 2[(3i_2 - n)\ mod\ 2] - 3[(-2i_2 + 21)\ mod\ 4])/4 \\ i_2 = (21 - (-2i_2 + 21)\ mod\ 4)/2 \\ \{n \leq (121 - 7[(-2i_2 + 21)\ mod\ 4] - 4[(3i_2 - n)\ mod\ 2])/4 \end{cases}$$

As the period of (ab) along i_2 is 2, instantiating i_2 will lead to two vertices:

(ab)	$\begin{pmatrix} (-2n+63-\left[\begin{matrix}3&5\\11&9\end{matrix}\right]_{i_2,n})/4 \\ (21-\left[\begin{matrix}1\\3\end{matrix}\right]_{i_2})/2 \end{pmatrix}$ $n \leq (\left[\begin{matrix}57&55\\48&50\end{matrix}\right]_{i_2,n})/2$
(ac)	$\begin{pmatrix} (-n+39+\left[\begin{matrix}3&32&33&20&21&8&9&38&39&26&27&14&15&2\\17&18&5&6&35&36&23&24&11&12&-1&0&29&30\end{matrix}\right]_{i_2,n})/14 \\ (2n+13+\left[\begin{matrix}1&13&11&9&7&5&3&15&13&11&9&7&5&3\\8&6&4&2&14&12&10&8&6&4&2&0&12&10\end{matrix}\right]_{i_2,n})/7 \end{pmatrix}$ $n \leq (121-\left[\begin{matrix}2&26&22&18&14&10&6&30&26&22&18&14&10&6\\16&12&8&4&28&24&20&16&12&8&4&0&24&20\end{matrix}\right]_{i_2,n})/4$
(ca)	$\begin{pmatrix} (-n+39+\left[\begin{matrix}3&4&5&6&7&8&9&-4&-3&-2&-1&0&1&2\\-11&-10&-9&-8&-7&-6&-5&-4&-3&-2&-1&0&-13&-12\\3&-10&-9&-8&-7&-6&-5&-4&-3&-2&-1&0&1&2\\3&4&5&6&-7&-6&-5&-4&-3&-2&-1&0&1&2\end{matrix}\right]_{i_2,n})/14 \\ (2n+13+\left[\begin{matrix}15&13&11&9&7&5&3&29&27&25&23&21&19&17\\22&20&18&16&14&12&10&8&6&4&2&0&26&24\\1&27&25&23&21&19&17&15&13&11&9&7&5&3\\8&6&4&2&28&26&24&22&20&18&16&14&12&10\end{matrix}\right]_{i_2,n})/7 \end{pmatrix}$ $n \leq (121-\left[\begin{matrix}15&13&11&9&7&5&3&29&27&25&23&21&19&17\\22&20&18&16&14&12&10&8&6&4&2&0&26&24\\1&27&25&23&21&19&17&15&13&11&9&7&5&3\\8&6&4&2&28&26&24&22&20&18&16&14&12&10\end{matrix}\right]_{i_2,n})/4$
(cb)	$\begin{pmatrix} \left[\begin{matrix}2&1&1&2\end{matrix}\right]_{i_2} \\ \left[\begin{matrix}8&9&10&7\end{matrix}\right]_{i_2} \end{pmatrix}$ $n \leq \left[\begin{matrix}20&25&28&17\end{matrix}\right]_{i_2}$

Figure 2.4: Explicit coordinates and validity domains of the pseudo-vertices of P_3.

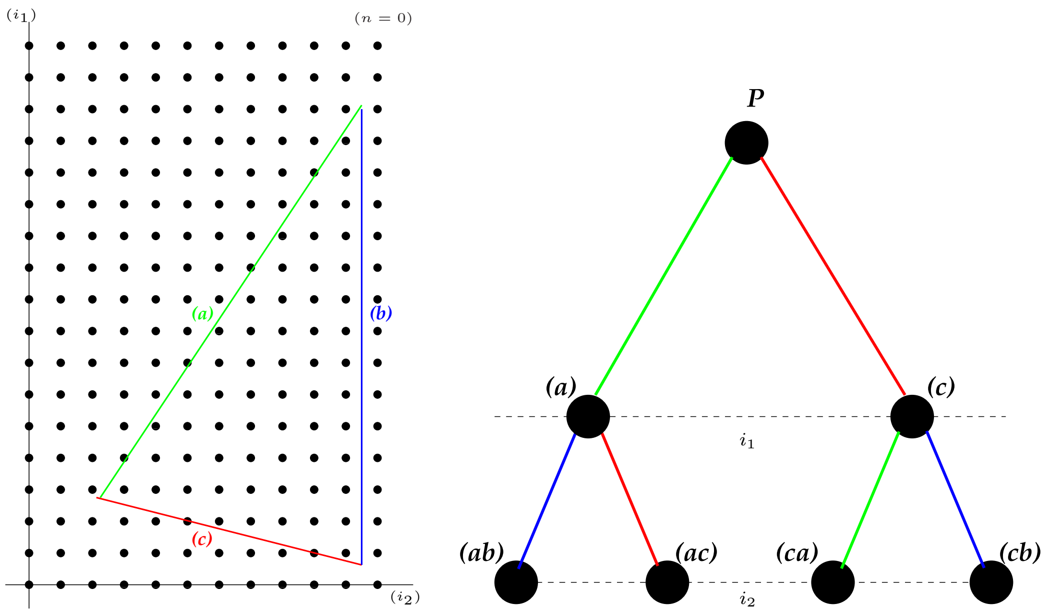

Figure 2.5: The faces of P_3 and its pseudo-facets tree.

- for i_2 *mod* $2 = 0$, we can take $i_2 = 0$:

$$\begin{cases} i_1 = (-2n + 63 - 2[(-n) \; mod \; 2] - 3[(21) \; mod \; 4])/4 \\ i_2 = (21 - (21) \; mod \; 4)/2 \\ \begin{cases} n \leq (121 - 7[(21) \; mod \; 4] - 4[(-n) \; mod \; 2])/4 \end{cases} \end{cases}$$

$$= \begin{cases} i_1 = (-n + 30 - [(-n) \; mod \; 2])/2 \\ i_2 = 10 \\ \begin{cases} n \leq (57 - 2[(-n) \; mod \; 2])/2 \end{cases} \end{cases}$$

- for i_2 *mod* $2 = 1$, we can take $i_2 = 659$:

$$\begin{cases} i_1 = (-2n + 63 - 2[(1977 - n) \; mod \; 2] - 3[(-1318 + 21) \; mod \; 4])/4 \\ i_2 = (21 - (-1318 + 21) \; mod \; 4)/2 \\ \begin{cases} 4n \leq (121 - 7[(-1318 + 21) \; mod \; 4] - 4[(1977 - n) \; mod \; 2])/4 \end{cases} \end{cases}$$

$$= \begin{cases} i_1 = (-n + 27 - [(1-n) \; mod \; 2])/2 \\ i_2 = 9 \\ \left\{ n \leq 25 - [(1-n) \; mod \; 2] \right. \end{cases}$$

Somehow, the variables are already instantiated in the explicit form. For instance, on figure 2.4, the coordinates of the nine extremal integer points (2 points for (ab), 2 for (ac), 4 for (ca) and 1 for (cb)) can be read directly from the explicit coordinates of the pseudo-vertices.

□

By opposition to cutting planes algorithms, this algorithm adds no constraint to the original problem, but constraints are replaced by periodic constraints. Also, 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 by computing a pseudo-facet n times.

2.4.1 Periodic validity domains

The algorithm presented in the previous section returns a set of pseudo-vertices. Each of them is expressed as the intersection between

- a set of n periodic hyperplanes, defining the coordinates of the pseudo-vertex,
- and the corresponding projected pseudo-facet, which is the projection of the pseudo-vertex into the parameter space.

This projection defines the values for which the pseudo-vertex belongs to P. It is similar to Loechner and Wilde's validity domains [55, 60] for rational vertices of a parametric polyhedron.

2.4.2 Computational complexity of the algorithm

We handle periodic numbers, which can have large periods if the coefficients of the constraints of P are large. An upper bound of the number of distinct polyhedra 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 respectively the number of variables and parameters[3]. But using the symbolic form of a periodic number avoids this exponential

[3] The precise number of polyhedra, corresponding to the period of $int(P)$, is computed in chapter 4.

complexity factor. Notice that this bound becomes polynomial, even with the explicit form, if the dimension of P and its number of parameters are fixed.

Computing all the pseudo-vertices of a n-dimensional simplex requires $(n+1)!$ computations of a pseudo-facet, which gives a lower complexity bound. An upper complexity bound can be derived by construction. As the order of variables to become integer is chosen once for the whole process, the number of operations is proportional to the number of distinct combinations of n inequalities among the m inequalities of P (without picking twice the same inequality), given by the well-known formula $\frac{m!}{(m-n)!(n!)}$. Both bounds are exponential in n, but polynomial for a fixed n. Finally, the complexity of finding the pseudo-vertices is then exponential, but polynomial for fixed n and p.

Anyway, the integer hull is the convex hull of the extremal points derived by instantiating the variables in the periodic part of the pseudo-vertices. This instantiation is exponential in function of n. Moreover, we know no algorithm that computes the convex hull of a set of extremal vertices under a symbolic form that contains nested modulo expressions. Thus, the only algorithm we can implement for now is exponential in function of n and p, and polynomial if n and p are fixed.

Nevertheless, the algorithm is highly parallel: the processing of subtrees of the pseudo-facet tree can be distributed, and the only data to be communicated is the pseudo-facet corresponding to the subtree root.

2.5 Generalization

2.5.1 Unbounded polyhedron

Unbounded polyhedra can be used to model nested *while* and *for* loops with the polytope model, as for instance in [23]. Nemhauser and Wolsey have shown in [69] (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 $conv(v^k)$.

Thus, computing the integer hull of an unbounded polyhedron P amounts to compute the integer hull of the convex hull of its extreme vertices.

2.5.2 Polyhedron with implicit equalities

The geometric dimension of a polyhedron P defined in a n-dimensional variable space by a non-redundant system of e equalities $E(I, N) = 0$ and e' inequalities is given by $d(P) = n - e$. A polyhedron is full-dimensional if $d(P) = n$, so if P has no equalities. We see in next section that the equalities of P must be handled in a specific manner. Following the terminology of Schrijver, the equalities can be classified into *explicit*, which are plain equalities in the definition of P, and *implicit* equalities, which are sets of inequalities defining P that are equivalent to a equalities. Once identified, implicit equalities can be transformed into explicit equalities. When some inequalities can be summed up into an (implicit) equality, we say that these inequalities *contribute* to an implicit equality.

In the parametric case, a little subtlety appears, as some inequalities can contribute to an implicit equality for only some values of the parameters. Along with the inequalities contributing to an implicit equality, we must then identify the values of the parameters for which the contributing inequalities are actually equivalent to an equality. We call these values the *contribution domain* of the inequalities to the implicit equality. Naturally, we are only interested in the values of the parameters that belong to the definition domain of P.

Example 2.12. The parametric polyhedron of variables (i, j, k) and parameterized by n:

$$P = \begin{cases} i \geq 0 \\ j \geq 0 \\ 2i - j + n \geq 0 \\ -2i + j - 3 \geq 0 \\ k = n \end{cases}$$

has one explicit equality, $k = n$. Besides, the two last inequalities contribute to one implicit equality for $n = 3$, as we have then:

$$0 \leq 2i - j + 3 \leq 0 \Leftrightarrow 2i - j + 3 = 0.$$

Hence, P is 2-dimensional for $n < 3$ and 1-dimensional for $n = 3$. The contribution domain of the two last inequalities to the implicit equality $2i - j + 3 = 0$ is $\{n = 3\}$.

□

The case where P has explicit equalities is treated in next section. We can assume for now that P is only defined by a set of non-redundant inequalities. Let K be a cone made of n inequalities of P. K can be written as:

$$AI + BN + C \geq 0,$$

where A, B and C are respectively $n \times n$, $n \times p$ and $n \times 1$ integer matrices.

Consider the system of inequalities $AI \geq 0$, which represents a cone K' centered at the origin. According to Schrijver ([76],chap. 8), the number of implicit equalities is $n - r$, where r is the row-rank of A. The row-rank of A is less than n if and only if some rows of A can be written as a linear combination of some other rows of A. This means that, when there is an implicit equality in K', one of the contributing inequalities can be written as a linear combination of the others. The number of implicit equalities is then given by the number of rows of A that can be eliminated by row eliminations, i.e. $n - r$. Let $UA = \begin{pmatrix} H \\ 0 \end{pmatrix}$ be the right Hermite normal form of A. The number of rows of H is the row-rank of A. The last rows with coefficients equal to zero correspond to inequalities that have been eliminated by (unimodular) row operations. The other rows that have been used for eliminating one of these last rows contribute to the same implcit equality. Since the k^{th} row of U defines the row operations performed on A to obtain the k^{th} row of $\begin{pmatrix} H \\ 0 \end{pmatrix}$, the $n - r$ last rows of U define the contributing inequalities for a given implicit equality: the j^{th} row of A (inequality of K) contributes to the implicit equality corresponding to the k^{th} row of UA if and only if U_{kj} is non-zero. For each implicit equality, we introduce a distinction among the contributing rows/inequalities:

- the k^{th} row is *eliminated* if the k^{th} row of UA is made of zeros,
- the k^{th} row is *remaining* if the k^{th} row of UA has non-zero coefficients.

Example 2.13. Let K be the set of three inequalities, where (i, j, k) are variables and

(n, m, l) are parameters:

$$K = \begin{cases} -2i + 3j - k - n \geq 0 \\ 2i - 3j + k - m \geq 0 \\ -2i + 3j - k - 2l \geq 0 \end{cases}$$

We can extract A:

$$A = \begin{pmatrix} -2 & 3 & -1 \\ 2 & -3 & 1 \\ -2 & 3 & -1 \end{pmatrix},$$

whose right Hermite normal form is:

$$UA = \begin{pmatrix} -1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -2 & 3 & -1 \\ 2 & -3 & 1 \\ -2 & 3 & -1 \end{pmatrix} = \begin{pmatrix} 2 & -3 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

The rank of A is one: there are two implicit equalities, which are materialized by the two last rows of the right-hand side matrix. Observe the second row of U: its first and second elements are non-zero, which means that the first and the second inequalities of K contribute to the implicit equality materialized by the second row of the right-hand side matrix. Similarly, one sees on the third row of U that the first and third inequalities of K contribute to the other implicit equality. Among the three contributing inequalities, only the first is remaining in UA, the second and the third being eliminated.

□

By definition, if the k^{th} contributing inequality of K is *eliminated*, the coefficients for the variables of the k^{th} row of the polyhedron resulting from the row-elimination U (let us call it UK) is made of zeros. But the coefficients for the parameters (and constants) are in general non-zero: they define equalities that are the condition on which the row elimination (and then the corresponding implicit equality) holds. In other words, the parameters (and constants) part in the k^{th} row of UK corresponding to the k^{th} *eliminated* row of UA define the contributing domain of the inequalities given in the k^{th} row of U. Besides, if the k^{th} contributing inequality of K is *remaining*, the k^{th} row of UK is the result of the elimination: it defines explicitly the implicit equality.

Example 2.14. Using K and U from the last example, we can apply the row elimination

U to K, giving:

$$U.K = \begin{cases} -2i + 3j - k - n = 0 \\ -n - m = 0 \\ n - 2l = 0 \end{cases} .$$

According to the third row of U:

- $n - m = 0$ is the contributing domain of $-2i + 3j - k - n \geq 0$ and $2i - 3j + k - m \geq 0$, which defines the equality $-2i + 3j - k - n = 0$,

- $n - 2l = 0$ is the contributing domain of $-2i + 3j - k - n \geq 0$ and $-2i + 3j - k - 2l \geq 0$, which (also) defines the equality $-2i + 3j - k - n = 0$.

□

As P is the intersection of all the possible K's derived from its defining inequalities, each distinct intersection of contribution domains may define a separate set of implicit inequalities. The definition domain of P can then be splitted into disjoint domains corresponding to each distinct set of contribution domains.

Intuitively, as the contributing domains are values of the parameters where some inequalities of P collapse into an equality, it seems that these values are the last values *until* P is empty. Therefore, we conjecture that the contributing domains of P are faces of its definition domain.

We will see in subsection 2.5.3 that the computation of the integer hull of a parametric polyhedron P depends on its defining equalities. The different contributing domains, where implicit equalities appear, must be treated separately.

2.5.3 Non-full-dimensional polyhedron

As our algorithm works with full-dimensional polytopes, we want to transform the rational non-full-dimensional parametric polyhedron P, defined by a non-redundant system of e equalities $E(I, N) = 0$ and e' inequalities, into a full-dimensional polyhedron. The e equalities define the parametric hyperplane $\epsilon(N) \in \mathbb{Q}^n$ on which the polyhedron lies (i.e., the affine hull of P), and the inequalities partition this hyperplane by half-spaces.

The equalities of a non-full-dimensional polyhedron make e variables dependent on the other ones: if $n - e$ variables are determined, the other variables are given by the e equalities. We can then use the equalities to eliminate e variables and project P in a

$(n-e)$-dimensional space. The resulting polyhedron is then full-dimensional and we can find its integer hull by the usual method. But, as we look for integer points, two problems appear:

- there must exist integer points in $\epsilon(N)$ for some integer values of N, or else $int(P)$ is always empty. Such values of N have to be determined.
- the e variables to be eliminated must be integer for any integer value of the remaining $n-e$ variables and of the parameters. If this is not the case, vertices that are integer in the $(n-e)$-dimensional space may not be integer when transformed back into the n-dimensional space.

Getting integer points in $\epsilon(N)$

The set of equalities in the definition of a parametric polyhedron $P(I,N)$ can be written:

$$A_e I + B_e N + C_e = 0, \tag{2.5}$$

In the general case of a non-full-dimensional polyhedron, P contains integer points only for some values of N, defined by an integer lattice. According to section 1.2, there is an integer point in $\epsilon(N)$ if and only if:

$$H_{A_e}^{-1}(BN + C) = D^{-1}H_{A_e}'^{-1}(BN + C) \in \mathbb{Z}^e \tag{2.6}$$

where $A_e = H_{A_e}U$ is the Hermite normal form of A_e, and D is a e-dimensional nonnegative diagonal matrix whose k^{th} diagonal element δ_k is the common denominator of the k^{th} row elements of the rational matrix $H_{A_e}^{-1}$. This necessary and sufficient condition can also be written:

$$[H_{A_e}'^{-1}(BN + C)] \; mod \; \delta = 0 \tag{2.7}$$

Or more shortly :

$$\Leftrightarrow [B'N + C'] \; mod \; \delta = 0 \tag{2.8}$$

where δ is the integer (e)-vector (δ_i). To solve this problem, first observe that solutions to:

$$B'N + C' = 0 \tag{2.9}$$

are also solutions to (2.8). The integer kernel K of B' is a basis for some integer solutions to (2.9): it is then a basis for some integer solutions to (2.8).

Now consider the i^{th} row of (2.8):

$$[B'_{i.}N + C'_i] \ mod \ \delta_i = 0, \tag{2.10}$$

where $B'_{i.}$ is the i^{th} row of B'. $(B'_{i.}N + C'_i) \ mod \ \delta_i$ is a periodic number, say of period $S_i = (s_{ij}), j \in [1..p]$. So if a given $N_0 \in \mathbb{Z}^p$ is solution to (2.10), any $N_1 \in \mathbb{Z}^p$ such that $N_1 \ mod \ S_i = N_0 \ mod \ S_i$ is also solution to (2.10): each equality of (2.8) yields some periodicity of the solution. Hence, if a given vector N_0 is solution to (2.8), any value N_1 such that $N_1 \ mod \ S = N_0 \ mod \ S$ is also solution to (2.8), where $S = (s_j)$, and $s_j = lcm_i(s_{ij})$, $j \in [1..p]$. s_j is the period along each parameter.

So the effect of the $mod \ \delta$ is to introduce periodicity of the solutions: the p-dimensional diagonal matrix $\mathcal{S} = (s_j)$ defines another basis for solutions to (2.8). Let $[G \ 0]$ be the p-dimensional (left) Hermite normal form of $\begin{pmatrix} K & \big| & \mathcal{S} \end{pmatrix}$ and N_0 a particular integer solution to (2.8). The general form of a solution to (2.8) is:

$$N = GN' + N_0 \tag{2.11}$$

If there is no N_0, there is no value of the parameters yielding an integer solution for I. The considered polytope contains then no integer point for any value of the parameters.

Finding a particular integer solution to (2.8) reduces to find an integer solution to $(B' \ mod \ \delta) \cdot N + D \cdot X = -C' \ mod \ \delta$, which can be written

$$\begin{pmatrix} B' \ mod \ \delta & D \end{pmatrix} \cdot \begin{pmatrix} N \\ X \end{pmatrix} = M \cdot \begin{pmatrix} N \\ X \end{pmatrix} = -C' \ mod \ \delta. \tag{2.12}$$

$N'_0 = H_M^{-1}.(-C' \ mod \ \delta)$ is integer if and only if there exists an integer solution to (2.12), where $M = [H_M \ 0] \cdot U_M$ is the Hermite normal form of M. The particular solution is then given by $N_0 = U_M^{-1} \cdot \begin{pmatrix} N'_0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ (see for instance [70]).

Equation (2.11) defines:

- a condition on N for the existence of an integer solution to equation (2.5), i.e., for the existence of an integer point in P,

- a compression of the parameters space (from N to N') so that there exists an integer point in P for any value of N'.

Notice that G is invertible (as S is invertible) and lower triangular. Moreover, as it is an integer compression, any integer value of N' corresponds to an integer value of N.

Example 2.15. Consider the following parametric rational polyhedron:

$$P_4 = \begin{cases} 2i + 4j - 3n + l + 5 = 0 \\ 3i - 9j + 2n - 4m - 2 = 0 \\ i + j - n + 2m \geq 0 \end{cases},$$

where i, j are the variables and n, m, l the integer parameters. Its affine hull is defined by

$$\epsilon_4(n, m, l) = \begin{cases} 2i + 4j - 3n + l + 5 = 0 \\ 3i - 9j + 2n - 4m - 2 = 0 \end{cases}.$$

We have:

$$H_{A_e} = \begin{pmatrix} 2 & 0 \\ 3 & 15 \end{pmatrix}, H_{A_e}^{-1} = \begin{pmatrix} \frac{1}{2} & 0 \\ \frac{-1}{10} & \frac{1}{15} \end{pmatrix}, \delta = \begin{pmatrix} 2 \\ 30 \end{pmatrix}, H'^{-1}_{A_e} = \begin{pmatrix} 1 & 0 \\ -3 & 2 \end{pmatrix}.$$

Hence, we have to solve:

$$\begin{cases} (-3n + l + 5) \ mod\ 2 = 0 \\ (13n - 8m - 3l - 19) \ mod\ 30 = 0 \end{cases}$$

The integer kernel of this system of equations, without the mod 's, is given by :

$$K = \begin{pmatrix} 2 \\ 1 \\ 6 \end{pmatrix}.$$

The period of the solution of the first equation is $\begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix}$ and it is $\begin{pmatrix} 30 \\ 15 \\ 10 \end{pmatrix}$ for the second equation. The global period of the solution is then given by their element-wise lcm:

$$S = \begin{pmatrix} 30 & 0 & 0 \\ 0 & 15 & 0 \\ 0 & 0 & 10 \end{pmatrix}$$

The basis of solutions is then given by (the non-null column-vectors of) the Hermite normal form of

$$(K|S) = \begin{pmatrix} 2 & 30 & 0 & 0 \\ 1 & 0 & 15 & 0 \\ 6 & 0 & 0 & 10 \end{pmatrix}, i.e., \left(\begin{array}{ccc|c} 2 & 0 & 0 & 0 \\ 1 & 15 & 0 & 0 \\ 6 & 0 & 10 & 0 \end{array}\right).$$

We can compute a particular solution:

$$\begin{pmatrix} n \\ m \\ l \end{pmatrix} = \begin{pmatrix} 7 \\ 0 \\ 34 \end{pmatrix},$$

so the condition on the parameters for $\epsilon_4(n, m, l)$ to contain an integer point is:

$$\begin{pmatrix} n \\ m \\ l \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 15 & 0 \\ 6 & 0 & 10 \end{pmatrix} \begin{pmatrix} n' \\ m' \\ l' \end{pmatrix} + \begin{pmatrix} 7 \\ 0 \\ 34 \end{pmatrix}.$$

Taken as a variable substitution, which corresponds to a compression of the parameters space, it gives the following transformed polyhedron:

$$\begin{cases} 2i + 4j + 10l' + 18 = 0 \\ 3i - 9j - 60m' + 12 = 0 \end{cases} = \begin{cases} i + 2j + 5l' + 9 = 0 \\ i - 3j - 20m' + 4 = 0 \end{cases},$$

which contains an integer point for any integer values of (n', m', l').

$\square$

In this subsection, we have seen that it is possible to compress the parameter space in order to have integer points in the affine hull of the resulting polyhedron for any values of the new parameters. In next subsection, we try to eliminate some variables by ensuring that they will have an integer value for any integer value of the other (remaining) variables and of the parameters. This is done using the same compression technique.

Integer variable elimination

Let $E(I, N) = 0$ the system of e equalities defining (along with a system of e' inequalities) the polyhedron P. The integer solutions to $E(I, N) = 0$ span a sub-lattice of $\mathbb{Z}^n$ whose determinant can be greater that one. Looking for integer points in P is then unobvious: we can find, from the inequalities defining P, integer points that are not solution to $E(I, N) = 0$.

Example 2.16. Consider the following polyhedron:

$$P_{eq} = \begin{cases} -2i + 3j - 1 = 0 \\ 2j - 3 \geq 0 \end{cases}.$$

The pseudo-facet obtained from the inequality $2j - 3 \geq 0$ w.r.t. j is $j = 2$, for which there is no integer solution to $-2i + 3j - 1 = 0$.

□

Problems with non-full-dimensional polyhedra in a space $\mathcal{S}$ of dimension n are commonly handled by projecting the polyhedron into a space $\mathcal{S}'$ of dimension $n - e$. The equalities $E(I, N) = 0$ allow to do this, by eliminating e variables. The projected polyhedron P' is full-dimensional, so we can compute the vertices of its integer hull by the normal method. But we seek for the corresponding integer vertices of P, which are integer solutions to $E(I, N) = 0$. Let I_2 be the variables to be eliminated and I_1 the other (remaining) variables. For the vertices of P to be integer in $\mathcal{S}$, it is necessary that, for any integer value of I_1 and N, the equalities $E(I_1, I_2, N) = 0$ lead to integer values of I_2:

$$(N, I_1) \in \mathbb{Z}^{p+n-e}, E(I_1, I_2, N) = 0 \Rightarrow I_2 \in \mathbb{Z}^e \tag{2.13}$$

As the e equations of $E(I, N) = 0$ determine the value of I_2 for a given value of (I_1, N), an equivalent condition is that there exists an integer solution I_2 to $E(I_1, I_2, , N) = 0$ for any values of (I_1, N). This condition on (N, I_1) is similar to the condition on N for the existence of integer solutions in I to $E(I, N) = 0$, computed in last subsection. By taking I_2 as variables and (N, I_1) as parameters, we then compute a compression on (N, I_1) such that I_2 is integer:

$$\begin{pmatrix} N \\ I_1 \\ 1 \end{pmatrix} = \left(\begin{array}{c|c} \mathrm{G} & \begin{matrix} N_0 \\ I_{1,0} \end{matrix} \\ \hline 0 & 1 \end{array}\right) \cdot \begin{pmatrix} N' \\ I_1' \\ 1 \end{pmatrix}, \tag{2.14}$$

where G is a lower triangular invertible integer matrix.

Substituting (I_1, N) by (I_1', N') and eliminating I_2 allows to find points with integer I_1' coordinates, whose corresponding points in $\mathcal{S}$ given by $E(I_1', I_2, N') = 0$ have integer I_2 coordinates. Moreover, as G, N_0 and $I_{1,0}$ are integer, (I_1, N) is integer for any integer (I_1', N').

Example 2.17. Consider the non-full-dimensional polyhedron

$$P_{eq2} = \begin{cases} -2i + 3j - n + 1 = 0 \\ 2j + i + 2n - 3 \geq 0 \end{cases}$$

in the 2-dimensional space (i, j), parameterized by n. Assume we chose to eliminate j. The compression such that $(i, n) \in \mathbb{Z}^2 \Rightarrow j \in \mathbb{Z}$ is : $\begin{pmatrix} n \\ i \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 3 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} n' \\ i' \\ 1 \end{pmatrix}$, which gives the compressed polyhedron:

$$P_{eq2}(i', n') = \begin{cases} -2i' + j - n' = 0 \\ 3i' + 2j + 3n' - 1 \geq 0 \end{cases}.$$

Eliminating j in $P_{eq2}(i', n')$ gives the polyhedron $\{7i' + 5n' - 1 \geq 0\}$, which is a full-dimensional polyhedron of dimension 1. The derived pseudo-vertex is given by

$$7i' + 5n' - 1 - (7i' + 5n' - 1) \ mod \ 7 = 0$$

which gives

$$i' = \frac{-5n' + 1 + (5n' - 1) \ mod \ 7}{7}.$$

As $i = n' + 3i'$ and $n' = n - 1$, we have:

$$i = \frac{-8n + 11 + 3(5n - 6) \ mod \ 7}{7} = \frac{-8n + 11 + \begin{bmatrix} 3 & 18 & 12 & 6 & 0 & 15 & 3 \end{bmatrix}_n}{7}$$

and, as $-2i' + j - n' = 0$,

$$j = \frac{-3n' + 2 + 2(5n' - 1) \ mod \ 7}{7} = \frac{-3n + 5 + \begin{bmatrix} 2 & 12 & 8 & 4 & 0 & 10 & 6 \end{bmatrix}_n}{7}$$

As expected, the resulting pseudo-vertex has integer coordinates. □

Notice that, as G is lower triangular, the computed compression (2.14) can be seen as a pair of compressions:

- the first p rows are a pure parameter space compression: N is a function of N' only.
- the other rows are a parametric compression: I_1 depends on I'_1 and on N'.

The parameter space compression corresponds to the condition on N for an integer point to exist in the affine hull of P. That is to say, it is (equivalent to) the parameter space compression computed in last section. Both compressions are then computed at the same time by ordering N and I_1 properly, as in (2.14). Notice that the parametric integer compression is non-orthogonal: it is not just a scaling. In the transformed space, the directions associated to the indices may be different than in the original space. In next section, we propose a way to compress I_1 and N so that I_2 is integer, but by using an orthogonal compression.

Constrained integer variable elimination

We may wish that the basis vectors for I'_1 have the same direction as for I_1. For instance, the lexicographic order is unchanged if I'_1 is obtained by a combination of an orthogonal scaling and a translation of I_1. In this case, the integer lexicographic extremum for (I_1, I_2) in P is obtained directly from the integer lexicographic extremum for I'_1 in P' using $E(I_1, I_2, N) = 0$.

A non-orthogonal compression from (N, I_1) to (N', I'_1) is given by equation (2.14). We use it to compute an orthogonal compression. Consider the $(p+k)^{th}$ row of equation (2.14), with $k \in [1, n]$:

$$i_k = G_{(p+k),1}n'_1 + \cdots + G_{(p+k),p}n'_p + G_{(p+k),(p+1)}i'_1 + \cdots + G_{(p+k),(p+k)}i'_k + i_{1,0,k}.$$

Let

$$i''_k = i'_k + \lfloor (G_{(p+k),(p+1)}i'_1 + \cdots + G_{(p+k),(p+k-1)}i'_{k-1} + i_{1,0,k})/G_{(p+k),(p+k)} \rfloor \tag{2.15}$$

We have:

$$i_k = G_{(p+k),1}n'_1 + \cdots + G_{(p+k),p}n'_p + G_{(p+k),(p+k)}i''_k + C_k,$$

where $C_k = (G_{(p+k),(p+1)}i'_1 + \cdots + G_{(p+k),(p+k-1)}i'_{k-1} + i_{1,0,k}) \ mod \ G_{(p+k),(p+k)}$. This relation between i_k and i''_k is a parametric scaling combined with a translation. This translation is periodic as C_k is periodic.

Notice that, according to (2.15), we have:

$$i''_k \in \mathbb{Z} \Leftrightarrow i'_k \in \mathbb{Z} \tag{2.16}$$

We would like to express I_1 as a function of $I_1'' = (i_k'')$, but C_k still depends on the i's. As G is lower triangular, i_k' depends on $(i_1' \cdots i_{k-1}')$ and on N'. Hence i_1' depends only on i_1'' and N, and transitively i_k' (as well as C_k) can be expressed as a function of $(i_1'', \cdots, i_{k-1}'')$ and N. Finally, the $(p+k)^{th}$ row of equation (2.13) can be written:

$$i_k = G_{p+k,0}n_0' + \cdots + G_{p+k,p}n_p' + G_{p+k,p+k}i_k'' + C_k'', \tag{2.17}$$

where C_k'' is C_k expressed as a function of I_1''.

Geometrically, the resulting transformation from I_1 to $I_1^{''}$ is the combination of an orthogonal scaling and a periodic parameterized translation. We call it naturally a *periodic parametric orthogonal compression*.

Example 2.18. Assume that the following condition holds on variables i, j, k and parameter n for some other variables to be eliminated:

$$\begin{pmatrix} n \\ i \\ j \\ k \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 2 \\ 2 & 3 & 0 & 0 & 4 \\ 1 & 1 & 2 & 0 & 2 \\ 2 & 2 & 3 & 4 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} n' \\ i' \\ j' \\ k' \\ 1 \end{pmatrix} \tag{2.18}$$

We have:

$$i'' = i' + \lfloor \frac{4}{3} \rfloor = i' + 1$$

so i' can be substituted in the second equation of system (2.18):

$$i = 2n' + 3i' + 4 = 2n' + 3i'' + 1$$

So $C_1 = 1$. The same process can be applied to the third equation:

$$j'' = j' + \lfloor \frac{i' + 2}{2} \rfloor \Leftrightarrow 2j'' = 2j' + i' + 2 - (i' + 2) \ mod \ 2$$

Substituting i', we obtain:

$$2j'' = 2j' + i'' + 1 - (i'' + 1) \ mod \ 2$$

and the third equation becomes:

$$j = n' + 2j'' + (i' - 2) \ mod \ 2 = n' + 2j'' + (i'' + 1) \ mod \ 2$$

The substitution for variable k is:

$$k'' = k' + \lfloor \frac{2i' + 3j' + 1}{4} \rfloor \Leftrightarrow 4k'' = 4k' + 2i' + 3j' + 1 - (2i' + 3j' + 1)\ mod\ 4$$

So the fourth equation becomes:

$$k = 2n' + 4k'' + (2i' + 3j' + 1)\ mod\ 4,$$

which gives, by substituting i' and j' by i and j:

$$\begin{aligned} k &= 2n' + 4k'' + (2i'' + 3j'' - 2 + \frac{-3i'' - 3 + 3(i'' + 1)\ mod\ 2}{2})\ mod\ 4 \\ &= 2n' + 4k'' + \frac{(i'' + 6j'' - 7 + 3(i'' + 1)\ mod\ 2)\ mod\ 8}{2} \\ &= 2n' + 4k'' + \begin{bmatrix} 2 & 1 & 3 & 2 & 0 & 3 & 1 & 0 \\ 1 & 0 & 2 & 1 & 3 & 2 & 0 & 3 \\ 0 & 3 & 1 & 0 & 2 & 1 & 3 & 2 \\ 3 & 2 & 0 & 3 & 1 & 0 & 2 & 1 \end{bmatrix}_{j'',i''} \end{aligned}$$

The parameterized compression defined by equation (2.18) can then be written as a periodic parameterized orthogonal compression:

$$\begin{pmatrix} n \\ i \\ j \\ k \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 2 \\ 2 & 3 & 0 & 0 & 1 \\ 1 & 0 & 2 & 0 & (i'' + 1)\ mod\ 2 \\ 2 & 0 & 0 & 4 & \frac{(i''+6j''-7+3(i''+1)\ mod\ 2)\ mod\ 8}{2} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} n' \\ i'' \\ j'' \\ k'' \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 & 2 \\ 2 & 3 & 0 & 0 & 1 \\ 1 & 0 & 2 & 0 & \begin{bmatrix} 1 & 0 \end{bmatrix}_{i''} \\ 2 & 0 & 0 & 4 & \begin{bmatrix} 2 & 1 & 3 & 2 & 0 & 3 & 1 & 0 \\ 1 & 0 & 2 & 1 & 3 & 2 & 0 & 3 \\ 0 & 3 & 1 & 0 & 2 & 1 & 3 & 2 \\ 3 & 2 & 0 & 3 & 1 & 0 & 2 & 1 \end{bmatrix}_{j'',i''} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} n' \\ i'' \\ j'' \\ k'' \\ 1 \end{pmatrix}$$

$\square$

2.5.4 Polyhedron with multiple validity domains

In the general case of a parameterized polyhedron, the computed extremal integer points will actually belong to the polyhedron P for only some values of the parameters. Loechner and Wilde give in [55, 60] an algorithm to compute the values of parameters for which a given rational vertex belongs to a rational polyhedron. This algorithm first computes the vertices of P and their projection into the parameters space (which is the validity domain of the vertex). The parameters space is then partitioned into polyhedral domains in which a given set of vertices belong to P, by a series of intersection. This method can be adapted quite directly to compute the values of the parameters associated to a given expression of the integer hull of P. Indeed, we can partition the parameter space into periodic polyhedra in which a given set of extremal integer points belong to P. However, we must separate the domains where implicit equalities appear, giving a beforehand partition. This is translated into the following algorithm:

```
get_integer_hull(Polyhedron P) {
  V, H: empty lists of periodic polyhedra
  E(I,N) = e explicit equations of P
  ∪(D_j, P_j) = Partition of the definition domain of P according to its contribution domains,
               and the corresponding expression of P
  for each (D_j, P_j)
    replace the implicit equalities of P_j by f explicit equalities
    compress the parameters and n-(e+f) variables according to E(I,N):
        (I,N) → (I', N')
    Q_j(I',N') = projection of P_j(I', N') by eliminating e+f variables.
    H = empty list of periodic polyhedra
    PX = empty list of periodic polyhedra
    PX = get_pseudo_vertices(Q_j)
    (W, X) = lists of validity domains and
           the corresponding valid extremal points from PX
    for each (W_i, X_i) in (W, X)
      H_i = convex hull of the pseudo-vertices X_i
      add (W_i, H_i) to (V, H)
    endfor
  endfor
  return (V, H)
}
```

Loechner-Wilde validity partitions the definition domain of P into adjacent polyhedral domains: these domains form a polyhedral complex[4]. This property, which does not

[4]This complex is known to be the *chamber complex* of P w.r.t. the projection into the parameters space

influence the algorithm, is not always true for the validity domains of the extremal integer points.

2.6 Convex hull of the integer points

To have an explicit form of the integer hull, the convex hull of the integer points must be computed. A convex hull algorithm must be run on a set of parametric points. To our knowledge, no such algorithm exists. Still, an algorithm based on Chernikova's transformation [12, 34, 51] exists for doing the converse operation: translating the set of parametric constraints of a polyhedron into its parametric vertices. For the non-parametric case, this algorithm is used for computing the convex hull of a set of points, by using duality. Adapting this method to the parametric case will not be tried here. However, even Chernikova's transformation, known to be among the fastest, has an exponential complexity in varying dimension.

Then, it sounds relevant to consider the peculiarities of the integer extremal points we have computed, in order to reduce the computation time of their convex hull. First, it is important to notice that the extremal integer points we compute are not the vertices of $int(P)$, but a superset of them. Some of the extremal integer points may strictly belong to $int(P)$ for any values of the parameters. These points, which we call *false integer vertices*, are then superfluous. They shall be eliminated, so that the integer hull can be computed directly using a (exponential) parametric convex hull algorithm. This issue is treated in next subsection. Then, a parametric convex hull algorithm, tailor-made to the case of our integer extremal points, is proposed in subsection 2.6.2.

2.6.1 Cleaning up

In the following discussion, each periodic value of the parameters is considered separately. Also, we consider the different validity domains separately. Then, in the considered values, the computed extremal vertices have a fixed affine and non-periodic expression. Let $P = \bigcap_q cone(v_q)$ be the decomposition of P into cones. As $int(P) = \bigcap_q int(cone(v_q))$, the vertices of $int(P)$ can be derived from each cone of P independently from the other cones. Hence, the false integer vertices can be eliminated cone-wise.

[77].

The problem we focus on can be stated as: Let v be a vertex of P, and v_{ext} the set of corresponding extremal integer points, i.e., the integer points derived by using the constraints containing v. Compute the set of integer points $v_{int} \in v_{ext}$ which are vertices of $int(P)$.

A first approach comes from noticing that:

- the computed set of extremal integer points depends on the (chosen) order in which the variables become integer,
- the vertices of $int(P)$ are extremal integer points for any chosen order.

Let A and B be sets of extremal integer points computed with the respective orders o_A and o_B. The points in $(A \cup B) \setminus (A \cap B)$ are not vertices of $int(P)$. It is then possible to eliminate false integer vertices by computing the extremal integer points for all the permutations of the variables and keeping only the points that are common to all the computations. This method may eliminate most false integer vertices, but not all of them in the general case. Moreover, it multiplies the computation complexity by $n!$, the number of permutations among the n variables.

Alternatively, we can rely on particular extremal integer points which are, by construction, close to the borders of P. These points are likely to be vertices of the integer hull: their convex hull is likely to contain other extremal points, which can then be eliminated. Consider a pseudo-vertex under its triangular form (as in section 2.4). The first row defines a periodic hyperplane which is parallel to a $(n-1)$-dimensional face of P. The periodic number of this row can then be seen as a distance between the face of P and the value of the periodic hyperplane. The second row can be read in a similar way: it corresponds to a periodic hyperplane parallel to the $(n-2)$-dimensional face of the projection of the $(n-1)$-dimensional facet mentioned above. The value of the periodic number can also be seen as a distance between both of them, and so on until the n^{th} row, whose periodic number is a distance between an extremal integer point and the corresponding rational vertex of P. Let us define this distance more formally:

Definition 3. *Let v be a vertex and u be an integer extremal point derived from a pseudo-vertex w corresponding to v. $h_k(u)$ is called the k^{th}* hyperplane distance *between u and v. It is the absolute value of the periodic number of the k^{th} row of w, where w is of triangular form.*

Following this idea of distance, we can define an order on the extremal integer points derived from the same vertex:

Definition 4. *Let u_1, u_2 be two extremal integer points derived from a vertex v. u_1 is said* closer to v than u_2 *if and only if:*

$$\exists k \in [1..n] : h_1(u_1) = h_1(u_2), h_2(u_1) = h_2(u_2) \dots h_{k-1}(u_1) = h_{k-1}(u_2), h_k(u_1) < h_k(u_2).$$

Notice that, due to the absolute value, this order is not total. For each pseudo-vertex, we can then define a (small set of) *closest integer extremal points* for each value of the parameters (modulo their period along the parameters). The maximum number of closest integer extremal points for a pseudo-vertex is bounded by 2^{n-1}, as there are only two possible minima for each row except for the first, where the periodic number can only be non-positive.

A heuristic for quickly eliminating some of the false integer vertices is based on the following conjecture:

Conjecture 1. *At least one of the closest integer extremal points of a pseudo-vertex is a vertex of $int(P)$.*

The following algorithm is then expected to remove a significant part of the false integer vertices (for each value of the parameters):

```
v: a vertex
Among the closest integer extremal points of the pseudo-vertices
 derived from v, pick n points: c_i, i ∈ [1..n].
Let u_i = c_i − v.
eliminate all the extremal integer points p such that
                    p − v = ∑(i = 1 to n)α_iu_i
with:
α_i ≥ 0
and ∑(i = 1 to n)α_i ≥ 1,
except the points c_i themselves.
```

Notice that the coordinates $(c_i - v)$ are easily available from the coordinates of the pseudo-vertex, which only differ from the coordinates of v by a periodic number.

It can be decided to repeat this elimination phase with different combinations of the closest extremal points. A stopping criterion has then to be set, which can be for instance the ratio between the false integer points that have been eliminated and the number of points before the elimination.

At this point, some false integer vertices still remain, and they have to be removed by computing the convex hull of all the remaining integer extremal points.

2.6.2 A tailored convex hull algorithm

First, we can take advantage of the fact that, for any extremal integer point c corresponding to a vertex v_q, the value of $c' = c - v$ is a constant. Working cone-wise, we can then run the non-parametric dual Chernikova transformation to obtain the convex hull of all the extremal integer points corresponding to a vertex. Actually, we must compute the convex hull of the extremal points *plus* the rays of $cone(v_q)$ to obtain $int(cone(v_q))$. The resulting constraints correspond to values of c', i.e., they are *translated* by v. Let $AI' + b \geq 0$ be one of the resulting constraints, computed from the points c'. Coming back to the constraints in the original coordinates I is straightforward:

$$I' = I - v \Rightarrow AI' + b = AI + b - Av.$$

The constraint is then given by $AI + b - Av$. It is naturally parametric as v is an affine function of the parameters.

Finally, as $int(P) = \bigcap_q int(cone(v_q))$, the integer hull is obtained by intersecting the cones (i.e., by putting together the constraints of the different cones). The overhead due to the parametric character of P is then small at this step of the integer hull computation.

Another question raises about the computation of the integer hull of P: when computing constraints from parametric vertices in general, one can find non-linear constraints, especially constraints whose coefficients depend on the parameters. In our case, the extremal integer points c corresponding to a vertex v are such that $c - v$ is constant (periodwise). Moreover, the vertices v are defined by the intersection of linear constraints (the constraints of P). These two elements are sufficient to state that the computed constraints cannot have coefficients that depend on parameters.

2.7 A remark on Ehrhart polynomials

The periodic character of the integer points of a rational parametric polyhedron has initially been pointed out by Ehrhart with his Ehrhart polynomials. The integer hull of the rational polyhedron P is the convex hull of periodic integer points: it is a periodic polyhedron with (periodic) integer vertices. It is periodic along the parameters. Ehrhart [26] has shown that the number of integer points in a rational parametric polyhedron (i.e., its Ehrhart polynomial) is non-periodic if all its vertices are integer. It is obvious that the Ehrhart polynomial of P equals the Ehrhart polynomial of $int(P)$. By definition, the periodic polyhedron $int(P)$ is represented by a set of polyhedra $\{M_I\}$ with integer vertices, each polyhedron defining $int(P)$ on a given integer (definition) lattice in the parameter space. The Ehrhart polynomial of M_I is then non-periodic, i.e. it is a (regular) polynomial.

We can deduce a new way to compute the Ehrhart polynomial $\mathcal{E}$ of P, by computing the Ehrhart polynomial of every M_I, each resulting non-periodic polynomial defining $\mathcal{E}$ on the integer lattice defining M_I. We do not discuss this algorithm, as we do not believe it is faster than current algorithms. However, it is interesting to see that the period of $\mathcal{E}$ equals the period of $int(P)$. This result will be useful in chapter 4, where Ehrhart polynomials are worked on.

2.8 Conclusion

We have defined a new method for computing the integer hull of a rational parameterized polyhedron P, which characterizes the integer points in P. New geometric objects and operators have been introduced for this purpose, which all deal with the periodic character of the extremal integer points of P. These objects, included in the class of *periodics*, are a conceptual bridge between the polyhedral model and the Ehrhart polynomials. As such, they fit in the framework developed around the polyhedral library Polylib [84, 56, 70]. This is why the algorithm for computing the pseudo-vertices of a non-full-dimensional periodic polyhedron has been developed using Polylib. The generic class of periodics has also been developed, including iterators for scanning the different values taken by a periodic.

We argue that the geometric framework of periodic polyhedra also allows new ways to tackle all the known problems due to integer points in parametric rational polyhedra. We show this in the next chapters by dealing with well-known problems in the polytope model: parametric linear integer programming problems, problems that come when projecting a $\mathbb{Z}$-polyhedron, and issues around Ehrhart polynomials.

Chapter 3

An application to integer linear programming

In chapter 2, we have seen that the integer points of a rational parametric polyhedron P can be characterized by the integer hull of P. This integer hull is the convex hull of some integer extremal vertices, which can be computed with the help of pseudo-facets. For some applications using this approach, only some of the vertices of the integer hull are looked for: the problem is to find minimal or maximal integer solutions that respect the affine constraints of P w.r.t. a certain order. In the polytope model, the natural order corresponds to the execution order of the iterations of the modeled loop nest: it is the *lexicographic order*. Feautrier [29] has developed the only algorithm for finding the integer lexicographic extrema of a rational parametric polyhedron, in the frame of the polytope model. This algorithm has been implemented in the PIP/PIPLib software [33]. Since then, numerous applications have been found in loop nest analysis, optimization and parallelization, for instance in [24, 8, 9, 22]. This order is total and non-linear, but it can be decomposed into a hierarchy of partial linear orders. This is shown in next section. By the way, we are lead to look for the integer optimum w.r.t. a linear order, which is a well-known mathematical problem whose applications outrun the scope of this thesis.

Many techniques exist in the literature for finding optimal integer vertices w.r.t. a linear order. Following Nemhauser and Wolsey ([69] chap. II) and Aardal and Van Hoesel [2, 3], they can be classified as follows:

- *cutting plane* algorithms compute the integer hull of a rational polyhedron P by adding constraints that exclude its non-integer vertices, 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 Dantzig, Fulkerson and Johnson. Gomory showed an automatic way to produce valid cutting planes in [36], giving birth to an algorithm that provably terminates. Chvàtal extended this algorithm to bounded polyhedra with real coefficients [13]. Also did Schrijver [76], who tackled the case of an unbounded polyhedron as well.

- *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. Branch and bound, as well as cutting planes, first look for a rational optimum at each step. Nowadays, this optimum is generally found by using the simplex method [25] or, alternatively, the interior point method [46, 49].

- an algorithm for deciding if an integer solution exists (which is actually the original formulation of the integer linear programming problem) has been described by Lenstra [43]. It is the first polynomial-time algorithm for a fixed dimension. By enclosing a transformation of P between two balls (one inside of P and one outside), and by finding a basis with special properties (a *reduced basis*) for the corresponding transformed lattice, it is shown that if there is an integer point in P, then it belongs to a set of hyperplanes whose number is polynomially bounded for a fixed dimension. Grötschel, Lovász and Schrijver [38, 76] refine this algorithm in some way by bounding P using ellipsoïds instead of balls, and Lovász and Scarf [62], as well as Kannan, use different definitions (so-called *generalized*) of a reduced basis. The interested reader is invited to have a look at the clear survey of Aardal [1] on the use of basis reduction for integer linear programming.

- notice that the question of the existence of an integer point in a polyhedron can be answered for a parametric polyhedron. In [9], Boulet and Redon use Feautrier's PIP to determine if a parametric polyhedron P contains an integer point, by determining the existence of a lexicographic integer point in P. We can also count the number of points in a parametric polyhedron, i.e., compute the Ehrhart polynomial of P, and test if it is positive. However, in practice the applicability of this test is limited to polynomials of small dimension. Two algorithms for computing the Ehrhart polynomial of a parametric polyhedron exist at present: the first one is due to Clauss, Loechner and Wilde [15, 21, 18] and the second to Seghir et al [77] and Verdoolaege et al [83].

- algorithms based on Lagrangean relaxation give an approximate solution by integrating constraints into the objective function. Since we are interested in exact solutions, the Lagrangean relaxation is not suitable.
- *column generation*, which is the dual equivalent for cutting planes.

Finding an integer extremum w.r.t. a partial linear order and w.r.t. the lexicographic order, which is total and non-linear, seem to be distinct problems. Actually, they can be considered as particular cases of a *hierarchically linear order*, which is introduced in next section. The aim of this chapter is then to find an algorithm for computing the extremum integer point of P w.r.t. a hierarchically linear order on $\mathbb{Q}^n$. An algorithm for solving this problem for full-dimensional polyhedra is presented in section 3.2. Specific parts of this algorithm are developed in sections 3.3 and 3.4. Then, it is extended to non-full-dimensional polyhedra in section 3.5.

3.1 Framework

Let $\mathcal{V}$ be an ordered set of w integer vectors $(v_1, v_2, \ldots, v_w)$, which can be represented as the row-vectors of a matrix V. $\mathcal{V}$ defines an order upon $\mathbb{Q}^n$, noted $\prec_{\mathcal{V}}$ and defined by:

$$I \in \mathbb{Q}^n \prec_{\mathcal{V}} J \in \mathbb{Q}^n$$

$$\Leftrightarrow \exists q \in [1..w] : v_1.I = v_1.J, v_2.I = v_2.J \ldots v_{q-1}.I = v_{q-1}.J, v_q.I < v_q.J$$

This order is total on $\mathbb{Q}^n$ if the vectors of $\mathcal{V}$ span an n-dimensional space. Else, it is a partial order on $\mathbb{Q}^n$. Let X_0 be a nonempty subset of $\mathbb{Q}^n$ (in which we look for a maximum), and let X_q the set of elements of X_{q-1} that maximize $v_q.I$. Basically, the maximum in X_0 w.r.t. $\mathcal{V}$ is X_w. It is a point if X_0 is finite and if $\mathcal{V}$ defines a total order. In this case, if X_0 is a rational polyhedron, X_w is a vertex of X_0. If X_0 is the intersection of a polyhedron P with $\mathbb{Z}^n$, X_w is a vertex of the integer hull of P.

Notice that the classical problem of linear programming can be seen as finding a maximum in a polyhedron P w.r.t. an order $\mathcal{V}$ defined by only one vector ($w = 1$). Besides, the lexicographic order is defined by $\mathcal{V} = (e_1, e_2, \cdots e_n)$, where $e_k, k \in [1..n]$, is the k^{th} vector of the canonical basis of $\mathbb{Z}^n$. In other terms, the lexicographic order is represented by a matrix V equal to the n-dimensional identity matrix.

In this chapter, we try to compute the integer points in a parametric polyhedron P which are maximal w.r.t. a given order $\mathcal{V}$ represented by matrix V. The corresponding minimum is the maximum w.r.t the order represented by matrix $-V$.

Pseudo-facets allow to compute extremal integer points of a rational parametric polyhedron. A straightforward way to compute the integer maximum of P w.r.t. $\mathcal{V}$ would then compute its integer hull P' and find the rational maximum in P', for example by adapting the lexicographic dual simplex algorithm (see [29] or [69] chap. II section 4). But this approach implies the computation of all the integer vertices of P', which is costly and in general useless as some vertices can never be the maximum w.r.t. $\mathcal{V}$. So, as it is generally done in integer linear programming, we must compute only a subset of the vertices of P', containing the integer maximum of P w.r.t. $\mathcal{V}$. Instead of computing all the extremal integer points of P, we can then compute only those which may contain the integer maximum w.r.t. $\mathcal{V}$. It is the principle of the algorithm presented in next section.

3.2 Computing the integer maximum w.r.t. $\mathcal{V}$

We proceed in two steps: first, we look for the set of pseudo-facets (f) that may contain the maximum w.r.t. $\mathcal{V}$. Then, we derive the integer extremal vertices from (f) and take the maximum among them. We first assume that the considered rational parametric polyhedron P is full-dimensional. The non-full-dimensional extension is treated in section 3.5.

Basically, the algorithm for computing the maximum pseudo-facets w.r.t. the hierarchically linear order represented by $\mathcal{V} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_w \end{pmatrix} I$ tries to maximize $v_1 I$. Then, it maximizes $v_2 I$ if the maximum for $v_1 I$ is degenerate (i.e., if it is a pseudo-facet whose dimension is greater than 0), and so on until finding one or many pseudo-vertices or having a degenerate maximum for $v_w I$. The special case where there is no maximum pseudo-facet along the current vector corresponds to an infinite solution. The general form of the algorithm for finding the pseudo-facets that are maximal w.r.t. $\mathcal{V}$ is then given by *maximize(P, V, 1)*, where the function *maximize* is defined as following:

```
maximize(cf, V, q,k) {
input: cf, the periodic polyhedron to be worked on
V: the hierarchically linear order to be considered
q: integer, the current row-vector of V that we have to maximize
k: we want i_k to become integer
output: a set S of pseudo-facets

0- if (v_q = zero vector) then {
      (a) if (q < w) then maximize(cf, V, q + 1, k)
      (b) else return cf
   }
1- (else)  select the set of pseudo-facets F of cf w.r.t i_k
   that may maximize v_qI
2- if F is empty, return +∞
3- (else) if (q = w) then return F
4- (else) for each pseudo-facet f ∈ F w.r.t. i_k {
   (a) decompose f into its supporting hyperplane h
       and the corresponding projected pseudo-facet pf
   (b) compute V', the projection of V along i_k using h
   (c) return (h ∩  maximize(pf, V', q, k + 1))
   }
```

Each call of *maximize* computes pseudo-facets for which $v_q I$ is maximal, and in which one more variable is integer. The algorithm stops either when the pseudo-facets are 0-dimensional, or when there is no more function to maximize. As the dimension of P and the number of vectors of $\mathcal{V}$ are bounded, it terminates. Its complexity is exponential, but polynomially bounded for a fixed dimension. It is clear that the complexity of this algorithm is strongly dependent on the way we *select*, at step 1, the pseudo-facets that may maximize $v_q I$: the fewer pseudo-facets it selects, the faster the algorithm will terminate.

The ideal selection method would select only one pseudo-vertex, which would render a polynomial complexity even without fixing the dimension, provided the selection method is itself polynomial. Selection methods are discussed in next section. If the selection method is not *ideal*, the result of *maximize(P, $\mathcal{V}$, 1, 1)* is either a pseudo-facet along with $+\infty$ or a set of pseudo-facets. Extracting the solution (i.e. a set of vertices) from this set is discussed in section 3.4.

3.3 Selection methods

Let $I \in \mathbb{Z}^n$ be the variables and $N \in \mathbb{Z}^p$ the parameters of the current periodic polyhedron P for which we seek a maximum w.r.t. $\mathcal{V}$.

3.3.1 A simple selection criterion

Let $v \in \mathcal{V}$ the vector corresponding to the current function vI to be maximized. A first selection method is to select the (periodic) inequalities $(C) : AI + BN + c \geq 0$ for which $Av < 0$, where A, B are integer $(1 \times n)$ and $(1 \times p)$ matrices, and c is a periodic number. Let us see why such inequalities lead to the maximum w.r.t. vI. Let U be a $n \times n$ matrix whose column-vectors have the following properties:

- the first is collinear to v: $v' = \alpha v, \alpha \neq 0$,
- the others, u_k with $k \in [2..n]$ are orthogonal to v: $u_k.v = 0$,
- they form a basis of $\mathbb{Q}^n$.

The (rational) change of variables $I = UI'$ implies: $vI = vUI' = \alpha \|v\|^2 i'_1$.

The maximum for vI is then given by the maximum value for i'_1. We can write (C) as:

$$AUI' + BN + c \geq 0.$$

It is clear that if Av, the coefficient of i'_1, is negative, (C) defines an upper bound on i'_1, i.e., the maximum value for i'_1 may be derived from (C). Hence, the integer points I for which vI are maximum can belong to a pseudo-facet derived from (C).

Example 3.1. Let us make a sketch of the process to find the candidate pseudo-vertices with the following problem (also considered by Feautrier in [29]): compute the integer lexicographic maximum of

$$P_1 = \begin{cases} -i+m \geq 0 \\ i \geq 0 \\ j \geq 0 \\ -j+n \geq 0 \\ 2i+j-k = 0 \end{cases}$$

where k, n, m are the parameters. The order matrix is given by $V = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. 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 that appears with non-full-dimensional polyhedra, presented in section 2.5.3 does not occur here.

$$P_1 = \begin{cases} 2i+j-k = 0 \\ \begin{cases} -i+m \geq 0 \\ i \geq 0 \\ -2i+k \geq 0 \\ 2i-k+n \geq 0 \end{cases} \end{cases}$$

The candidate vertices belong to two max-pseudo-facets of P_1 w.r.t. i:

$$A = \begin{cases} 2i+j-k = 0 \\ \begin{cases} -i+m & = 0 \\ i & \geq 0 \\ -2i+k & \geq 0 \\ 2i-k+n & \geq 0 \end{cases} \end{cases} = \begin{cases} 2i+j-k = 0 \\ -i+m = 0 \\ \begin{cases} m & \geq 0 \\ k-2m & \geq 0 \\ -k+2m+n & \geq 0 \end{cases} \end{cases}$$

$$B = \begin{cases} 2i+j-k = 0 \\ \begin{cases} -i+m & \geq 0 \\ i & \geq 0 \\ -2i+k-(-2i+k)\ mod\ 2 & = 0 \\ 2i-k+n & \geq 0 \end{cases} \end{cases} = \begin{cases} 2i+j-k = 0 \\ -2i+k-\begin{bmatrix} 0 & 1 \end{bmatrix}_k = 0 \\ \begin{cases} -k+2m+\begin{bmatrix} 0 & 1 \end{bmatrix}_k & \geq 0 \\ k-\begin{bmatrix} 0 & 1 \end{bmatrix}_k & \geq 0 \\ n-\begin{bmatrix} 0 & 1 \end{bmatrix}_k & \geq 0 \end{cases} \end{cases}$$

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

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

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

□

In this example, the periodic numbers defining the found pseudo-vertices are not function of the variables (which is generally the case). Section 3.4 shows how to derive integer points from pseudo-vertices.

3.3.2 An ideal selection criterion?

Intuitively, selecting the inequality for which $A.v^T$ is minimal looks like a reasonable selection criterion, which would dramatically reduce the number of pseudo-facets to be selected at each step. Unfortunately, this intuition is not validated by a proof yet. Anyway, the parameter space would have to be split into domains for which an inequality minimizing $A.v^T$ is not redundant in P. Actually, this splitting work is already done when computing the validity domains of the different candidate maximal integer points, with the simple selection method. But this latter computation is clearly heavier than just splitting the parameters space before choosing the pseudo-facets.

3.4 From pseudo-facets to the solution

In this section, we assume that the result of the function *maximize(P, $\mathcal{V}$, 1)* is a set of pseudo-facets, each q-dimensional pseudo-facet being decomposed into a set of $n - q$ periodic hyperplanes and the corresponding projected pseudo-facet (it is projected in a space with q variables and p parameters). Two particular cases can appear:

- some of the pseudo-facets correspond to an infinite maximum (*maximize* returns ∞),
- all the pseudo-facets are pseudo-vertices (they are 0-dimensional).

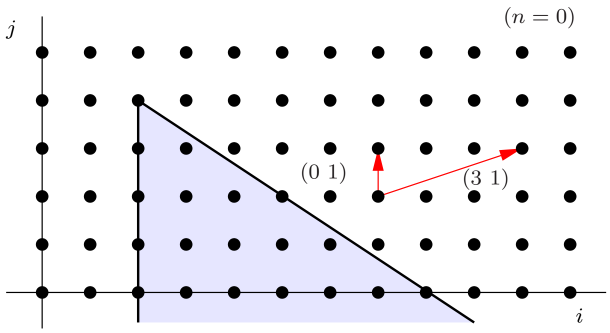

Figure 3.1: The polyhedron P_∞

3.4.1 Infinite solutions

In the case where the function maximize returns $+\infty$, the solution is infinite. However, as $+\infty$ results from the maximum w.r.t. $\mathcal{V}$ in a periodic polyhedron cf, we know that this infinite solution belongs to the affine hull of cf, which is more precise than a mere *infinity*. In the general case, cf is a q-dimensional pseudo-facet of P, so its affine hull is a q-dimensional periodic hyperplane.

Example 3.2. Assume that we look for the integer maximum w.r.t the order $\mathcal{V}$ given by $V = \begin{pmatrix} 3 & 1 \\ 0 & 1 \end{pmatrix}$ for the polyhedron P_∞ represented on figure 3.1.

$$P_\infty = \begin{cases} i - 2 \geq 0 \\ -2i - 3j + n + 16 \geq 0 \end{cases}$$

The first function to be maximized is $3i + j$: let us compute the pseudo-facet(s) of P_∞ corresponding to an upper bound of $3i + j$. They are derived from the inequalities $AI + BN + c \geq 0$ for which $A.(3\ 1)^T$ is negative. Only the second inequality is in this case. The integer maximum for $3i + j$ belongs then to the pseudo-facet:

$$\begin{cases} -2i - 3j + n + 16 - (-3j + n + 16)\ mod\ 2 = 0 \\ i - 2 \geq 0 \end{cases} = \begin{cases} 2i = -3j + n + 16 - (j + n)\ mod\ 2 \\ \{-3j + n + 12 - (j + n)\ mod\ 2 \geq 0 \end{cases}$$

The projection of $3i + j$ along i using $2i = -3i + n + 16 - (j + n)\ mod\ 2$ is $-7j$. The new function to maximize is then $-j$ (put to a g.c.d. of 1). Now, there is no inequality

$AI + BN + c \geq 0$ for which $A.(0\ \ -1)^T$ is negative any more: the maximum value for the free variable j is $+\infty$. The solution can then be written:

$$max(P_\infty, \mathcal{V}) = \begin{cases} 2i = -3j + n + 16 - (j + n)\ mod\ 2 \\ j = +\infty \end{cases}$$

□

3.4.2 Set of pseudo-vertices

In the case where the returned pseudo-facets are all 0-dimensional, the maximum integer extremal points w.r.t. $\mathcal{V}$ must be computed for each value of the parameters (modulo the period along the parameters). The overall maximal extremal integer points can be computed by selecting the maximum integer extremal points w.r.t. $\mathcal{V}$ for each pseudo-vertex and then taking the maxima among them. According to section 2.6.1, the diagonal form of a pseudo-vertex u gives directly the coordinates of u in function of its corresponding vertex v (of P): $u = v - c$, where c is a periodic number. When finding the maximal integer extremal points corresponding to a given pseudo-vertex, comparing the different values of $v.c$ (instead of $v.u$) is then sufficient. But each integer extremal point has a different validity domain: the maximum is only the maximum inside its definition domain. Outside, one of the other integer extremal points may be the maximum, and so on. The solution is then given by the following algorithm :

for each pseudo-vertex u
X: the set of extremal integer points corresponding to u
(M, D): list of the maxima and their associated domain of parameters
$(M, D)\ =\ (\emptyset,\ \emptyset)$
D': the domain of parameters for which the maximum is determined
$D'\ =\ \emptyset$
while $X \neq\ \emptyset\ \ \{$
 select the set $x \subseteq\ X$ with the maximal value w.r.t. $\mathcal{V}$
 their validity domain is dv_x
 if $(dv\_x\ \setminus\ D')\ \neq\ \emptyset\ \{$
  add $(x,\ dv\_x\ \setminus\ D')$ to (M, D)
  $D'\ =\ D'\ \cup\ dv\_x$

```
  }
  X  =  X \ x
}
```

Notice that the inequalities of the validity domains only vary from one constraint, from one extremal integer point to another. Consider two extremal integer points u_1 and u_2. Also, consider one of the constraints of their validity domains, respectively $BN + c_1 \geq 0$ and $BN + c_2 \geq 0$: if $c1 \geq c_2$, then $BN + c_2 \geq 0 \Rightarrow BN + c_1 \geq 0$. This allows to test quickly if the validity domain of u_1 is included in the validity domain of u_2, by using only the c's.

Example 3.3. The pseudo-vertex of P_3 (from example 2.10) that is maximal w.r.t. the order $\mathcal{V}_3$ defined by $V = \begin{pmatrix} 1 & -1 \end{pmatrix}$ is (ab), whose diagonal form is:

$$\begin{cases} i = (-2n + 63 - \begin{bmatrix} 3 & 5 \\ 11 & 9 \end{bmatrix}_{i_2,n})/4 \\ j = (21 - \begin{bmatrix} 1 \\ 3 \end{bmatrix}_{i_2})/2 \\ \begin{cases} -2n + \begin{bmatrix} 57 & 55 \\ 48 & 50 \end{bmatrix}_{i_2,n} \geq 0 \end{cases} \end{cases}$$

we can focus on the periodic numbers:

$$c = \begin{pmatrix} -(\begin{bmatrix} 3 & 5 \\ 11 & 9 \end{bmatrix}_{i_2,n})/4 \\ -(\begin{bmatrix} 1 \\ 3 \end{bmatrix}_{i_2})/2 \end{pmatrix}$$

We have:

$$V.c = \begin{pmatrix} -(\begin{bmatrix} 1 & 3 \\ 5 & 3 \end{bmatrix}_{i_2,n})/4 \\ 0 \end{pmatrix} \quad for \quad \begin{cases} -2n + \begin{bmatrix} 57 & 55 \\ 48 & 50 \end{bmatrix}_{i_2,n} \geq 0 \end{cases}$$

For $n \; mod \; 2 = 0$, the maximum for $V.c$ is $\begin{pmatrix} -1/4 \\ 0 \end{pmatrix}$, which is reached for $i_2 \; mod \; 2 = 0$, i.e. for $\begin{pmatrix} i \\ j \end{pmatrix} = \begin{pmatrix} -(n/2) + 15 \\ 10 \end{pmatrix}$, with a validity domain of $-2n + 57 \geq 0$. This validity domain includes $-2n+48 \geq 0$, so the extremal integer point obtained with $i_2 \; mod \; 2 = 1$ is never the maximum.

For $n \; mod \; 2 = 1$, the maximum for $V.c$ is $\begin{pmatrix} -3/4 \\ 0 \end{pmatrix}$, which is reached for $i_2 \; mod \; 2 = 0$ and for $i_2 \; mod \; 2 = 1$, i.e. for $\begin{pmatrix} i \\ j \end{pmatrix} = \begin{pmatrix} -(n-1)/2+14 \\ 10 \end{pmatrix}$ and $\begin{pmatrix} i \\ j \end{pmatrix} = \begin{pmatrix} -(n-1)/2+13 \\ 9 \end{pmatrix}$, with a validity domain of $-2n+55 \geq 0$. The solution can then be written in a periodic manner:

$$\left\{ \begin{bmatrix} \begin{pmatrix} -n/2+15 \\ 10 \end{pmatrix} & \begin{pmatrix} -(n-1)/2+14 \\ 10 \end{pmatrix} \end{bmatrix}_n ; \begin{bmatrix} \begin{pmatrix} -n/2+15 \\ 10 \end{pmatrix} & \begin{pmatrix} -(n-1)/2+13 \\ 9 \end{pmatrix} \end{bmatrix}_n \right\}$$

respectively for

$$\left\{ -2n + \begin{bmatrix} 57 & 55 \end{bmatrix}_n \geq 0; -2n + \begin{bmatrix} 57 & 50 \end{bmatrix}_n \geq 0 \right\}.$$

□

3.4.3 General case

In the general case, function *maximize* returns a set of pseudo-facets of different dimensions. By comparison with the rational case, we assume that the maximal integer vertices w.r.t. a given order is given by only one pseudo-facet f. Then, any point I (even rational) of f gives, for each periodic value of the variables and the parameters, the maximum value for VI that can take an integer point of f. The maximum can then be determined as in the previous subsection, by evaluating VI.

Example 3.4. As nothing new specific to periodicity or to the parametric character of the computation is discussed in the current subsection, let us consider an example without parameters and whose solution is non-periodic. Let us find the integer maximum w.r.t. $\mathcal{V}_5$ with $V_5 = \begin{pmatrix} 1 & 4 & 1 \end{pmatrix}$ in

$$P_5 = \begin{cases} i+k \geq 0 \\ -i-k+2 \geq 0 \\ -i-j-k+3 \geq 0 \\ -i+k+2 \geq 0 \\ i-k+2 \geq 0 \end{cases}$$

P_5 is represented on figure 3.2. Let $v = (1 \; 4 \; 1)$. The inequalities $AI + c \geq 0$ for which

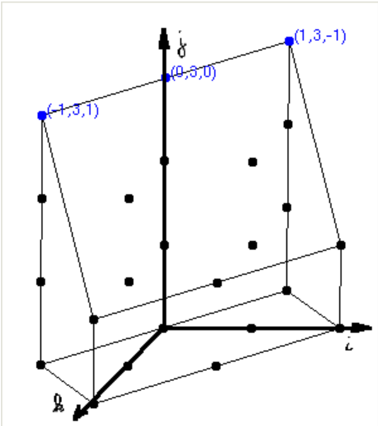

Figure 3.2: P_5 and its integer maxima w.r.t. $\mathcal{V}_5$

$A.v < 0$ are $-i-k+2 \geq 0$ and $-i-k-j+3 \geq 0$, giving respectively the following pseudo-facets pf_1 and pf_2:

$$pf_1 = \begin{cases} -i-k+2=0 \\ \begin{cases} 2 \geq 0 \\ -j+1 \geq 0 \\ 2k \geq 0 \\ -2k+4 \geq 0 \end{cases} \end{cases}, pf_2 = \begin{cases} -i-j-k+3=0 \\ \begin{cases} -j+3 \geq 0 \\ j-1 \geq 0 \\ j+2k-1 \geq 0 \\ -j-2k+5 \geq 0 \end{cases} \end{cases}.$$

On pf_1, $vI = i+4j+k = 4j+2$, so $v = (0\ 4\ 0)$, or $(0\ 1\ 0)$. The only inequality of pf_1 for which $A.v$ is negative is $-j+1 \geq 0$, so the pseudo-facet pf_{11} may contain the maximum w.r.t. $\mathcal{V}_5$:

$$pf_{11} = \begin{cases} -i-k+2=0 \\ -j+1=0 \\ \begin{cases} k \geq 0 \\ -k+2 \geq 0 \end{cases} \end{cases}.$$

On pf_{11}, $vI = j = 1$, so $v = (0\ 0\ 0)$ and we cannot go further. If pf_{11} is the maximum w.r.t. $\mathcal{V}_5$, it is a one-dimensional degenerate solution. We can then take any value of k to compute vI on pf_{11}:

$$vI = (1\ 4\ 1).((-k+2)\ 1\ k)^T = 6.$$

On pf_2, $vI = i + 4j + k = 3j + 3$, so $v = (0\ 3\ 0)$, or equivalently $v = (0\ 1\ 0)$. Two pseudo-facets of pf_2 can maximize vI:

$$pf_{21} = \begin{cases} -i-j-k+3=0 \\ -j+3=0 \\ \begin{cases} 2 \geq 0 \\ k+1 \geq 0 \\ -k+1 \geq 0 \end{cases} \end{cases}.$$

Again, $v = (0\ 0\ 0)$ on pf_{21}, so the solution is degenerate as pf_{21} is one-dimensional. We have:

$$vI = (1\ 4\ 1).(-k\ 3\ k)^T = 12.$$

Besides,

$$pf_{22} = \begin{cases} -i-j-k+3=0 \\ -j-2k+5=0 \\ \begin{cases} k-1 \geq 0 \\ -k+2 \geq 0 \\ 2 \geq 0 \end{cases} \end{cases}$$

On pf_{22}, $v = j = -2k + 5$, so $v = (0\ 0\ -1)$. The pseudo-facet that maximizes v in pf_{22} is given by:

$$pf_{221} = \begin{cases} -i-j-k+3=0 \\ -j-2k+5=0 \\ k-1=0 \\ \begin{cases} 2 \geq 0 \end{cases} \end{cases}$$

It is the 0-dimensional pseudo-facet $(-1\ 3\ 1)$, so if it is the solution, it is non-degenerate. For pf_{221}, we have:

$$vI = (1\ 4\ 1).(-1\ 3\ 1)^T = 12.$$

Noticing that pf_{221} belongs to pf_{21}, the solution is given by pf_{21}.

□

When the maximum for $v_k I$ in a periodic polyhedron cf is degenerate, it corresponds to an inequality $AI + BN + c \geq 0$ such that the change of variables $I = UI'$ defined in section 3.3 would give a constant upper bound on i'_1 (as $A.u_k = 0, k \in [2..n]$). As cf is convex, its value of $v_k I$ is then greater than the values for other pseudo-facets of cf. This allows to prune the branches of the candidate pseudo-facets tree: if a degenerate maximum for $v_k I$ is found in a q-dimensional pseudo-facet cf, the other $q-1$-dimensional candidate pseudo-facets derived from cf can be eliminated, as they do not contain a better solution. In example 3.4, the pseudo-facet pf_{22} would have been pruned.

3.5 Non-full-dimensional polyhedra

As we have seen in section 2.5.3, a non-full-dimensional polyhedron P defined by a set of e equalities (along with inequalities) can be transformed into a full-dimensional polyhedron, by compressing it into a polyhedron P' (so that its supporting lattice is $\mathbb{Z}^n$) and then projecting it into a space where it is full-dimensional, giving a polyhedron P''.

Our algorithm for finding an integer maximum w.r.t. a hierarchically linear order $\mathcal{V}$ is based on the construction of pseudo-facets, selected according to the scalar product between the row-vectors of the matrix V representing $\mathcal{V}$ and the normal vector of faces of P. As P is compressed, we must compute the hierarchically linear order $\mathcal{V}'$ whose integer maximum in P' corresponds to the integer maximum w.r.t. $\mathcal{V}$ in P. Hence, the selected inequalities of P' according to $\mathcal{V}'$ must correspond to the inequalities of P that would have been selected according to $\mathcal{V}$. Let U be the compression: an inequality of P written as $AI + BN + c \geq 0$ is compressed according to $I = UI'$. It is then compressed into the inequality $AUI' + BN + c \geq 0$, i.e., $A'I' + BN + C \geq 0$ with $A' = AU$. Let v a row-vector of V. When we look for the integer maximum for vI, we compute pseudo-facets by selecting the inequalities for which $A.v^T < 0$. Let $v'^T = U^{-1}.v^T$. We have

$$A.v^T = A'U^{-1}Uv'^T = A'v'^T.$$

In other words, an inequality of P' would be selected according to v if and only if the compressed inequality is selected according to $v' = v.(U^{-1})^T$. Finally, looking for the integer maximum in P w.r.t. $\mathcal{V}$ reduces then to looking for the integer maximum in P' w.r.t. the order $\mathcal{V}'$ whose representing matrix is $V' = V.(U^{-1})^T$.

Similarly, when P' is projected using the e equalities into a full-dimensional polyhedron P'', the order $\mathcal{V}'$ must be transformed accordingly, by using the e equalities to eliminate e variables in $V'I$, giving the order $\mathcal{V}''$.

Once the integer maximum in P'' w.r.t $\mathcal{V}''$ found, the solution to the original problem is obtained by computing the values of the e eliminated variables using the e equalities, and then by uncompressing the variables, using $I = UI'$.

3.6 An application: determining data holes

The polytope model has broadly shown its power as a model for loop nests analysis. Several analysis, optimization and parallelization methods are obtained by transforming parametric $\mathbb{Z}$-polyhedra (and thus their associated loop nest), for instance by:

- applying affine transformations, including scalings, skewings, symetries (as for instance in [53, 71, 45]),
- applying non-affine transformations, as for instance tiling, unrolling, linearizing, peeling (see for instance [86]),
- splitting into sub-($\mathbb{Z}$-)polyhedra (e.g. in [37, 67, 58]),

resulting in a set of parametric disjoint Z-polyhedra. But a noticeable weakness of the model shows up when the transforming function is non-invertible: the image of a $\mathbb{Z}$-polyhedron by such a function is, in general, not a $\mathbb{Z}$-polyhedron.

Again, from now on we will consider $\mathbb{Z}$-polyhedra whose lattice is $\mathbb{Z}^n$, as it is always possible to transform a $\mathbb{Z}$-polyhedron to this simpler form.

Example 3.5. The following example has been presented by Clauss [17]. It is a parametric version of an example initially presented by Ferrante *et al* [35] and also treated by Pugh [73].

```
for i = 1 to 8 do
  for j = 1 to p do
    a(6i+9j-7) = ...
```

The elements of the array a reached by this loop are defined by :

$$E = \{k \in \mathbb{Z} \mid \exists i, j, p \in \mathbb{Z}^3 : 1 \leq i \leq 8; 1 \leq j \leq p; k = 6i + 9j - 7\}$$

Although these points are defined by a set of affine rational constraints (*i.e.,* equalities and inequalities with integer coefficients), the set E of integer values of k, which is the image of a $\mathbb{Z}$-polyhedron by a non-invertible affine function, cannot be described by a $\mathbb{Z}$-polyhedron. E is the $\mathbb{Z}$-polyhedron $\{k = 3x + 2, 8 \leq k \leq 9p + 41, x \in \mathbb{Z}\}$ *minus* the two points $\{k = 11\}$ and $\{k = 9p + 38\}$. These two points that are not accessed by the loop nest will be called *(data) holes* from now on.

□

As seen in example 3.5, a non-invertible affine transformation can be written as a projection, by introducing a new variable (k in the example). Hence, we will use the term *projection* in the following. As remarked by Aardal in [1], Lenstra [43] and Lovasz and Scarf [62] based their integer linear programming algorithms on the idea that integer points are hard to find in *thin* polyhedra. The projection of a thin polyhedron P is likely to contain integer points which do not result from the projection of an integer point of P. Formally, we can say that the two applications: projection and intersection with a lattice of points, do not commute in general:

$$\Pi(P \cap L) = \Pi(P) \cap \Pi(L) \ is\ false,$$

where Π is a projection and L is a lattice of points. This observation is confirmed by Pugh, whose Omega test [72] identifies a *dark shadow* in the projection of P (which is itself called the *real shadow* of P). The integer points in the dark shadow, which is a sub-polyhedron of the real shadow, are mandatorily resulting from the projection of an integer point of P. The rest of the real shadow, which results from the projection of thin parts of P, may contain integer points that are not the projection of integer points of P (we call these points *holes*). In [72], the aim of Pugh is not to identify holes or the integer points that are not holes, but to tell if a polyhedron contains integer points. The problem Pugh focuses on is the existence of data dependences modeled by a polyhedron. Also, Feautrier's PIP, which computes the lexicographic extrema of a parametric polyhedron, was firstly aimed at computing dependence functions and (re-)scheduling loop nests [30]. As stressed later by Pugh [73] in the context of integer points enumeration, the problem of finding holes in the projection of a $\mathbb{Z}$-polyhedron can be solved by solving the problem of the existence of an integer point in a polyhedron.

This link between the two problems is used in [9] by Boulet and Redon. When pre-evaluating the communication volumes induced by a given alignment and distribution for a HPF program, they need to count only once the number of template locations to be read without communication (because they are mapped to the same processor as the written location) at a given loop iteration. As the distribution (which maps the template locations to the processors) is non-invertible, the authors use Feautrier's PIP to consider only one of the read template locations (the lexicographic minimum among them) per processor. This leads somehow to a one-to-one mapping between a processor and a read template location, which allows to count the number of iterations which do not generate communications for each template location. The total number of communications among template locations is then derived, using Ehrhart polynomials. In this frame, a processor which is in the image of the polyhedron defining the template locations, but on which no template location (defined by integer coordinates) is distributed, is a hole.

Until now, the unarguably appropriate way to count the computation and data volumes of a loop nest is by using Ehrhart polynomials [26]. It has been initially proposed by Clauss [17]. The same author uses Ehrhart polynomials to find holes in the projection of a $\mathbb{Z}$-polyhedron P, by determining the number of integer points of P that have an integer image in the projection of P. This number is an Ehrhart polynomial, whose integer roots correspond to the holes. The feasibility of this technique is then limited by the possibility of finding integer roots of multivariate polynomials. Counting the holes allows then to count the number of integer points in the projection of P.

We show in subsection 3.6.1 how to determine the exact set of values accessed by a loop nest through a non-invertible reference function. This result gives directly an alternative method, developed in subsection 3.6.2, for computing the number of integer points in the image of a $\mathbb{Z}$-polyhedron by a singular affine function.

3.6.1 Domain of existence of an integer point in a polyhedron

In the polytope model, the iterations of a loop nest correspond to the integer points of a $\mathbb{Z}$-polytope $P(I, N) : \mathcal{P}.\begin{pmatrix} I \\ N \\ 1 \end{pmatrix} \geq 0 \cap \mathbb{Z}^{n+p+1}$, while data accessed through (multi-dimensional) access functions are the image of these points by the corresponding mapping. This mapping can be represented by an access matrix. It has been shown [17] that when this matrix is non-invertible, the image is not a $\mathbb{Z}$-polyhedron in the general case, but a union of $\mathbb{Z}$-polyhedra. Consider a $m \times (n + p + 1)$ access matrix A, through which a loop with n indices and p parameters accesses a m-dimensional array. Let $X \in \mathbb{Z}^m$ be the coordinates of an accessed array element. The access of an element X by an iteration I of P is solution to:

$$\{\mathcal{P}.\begin{pmatrix} I \\ N \\ 1 \end{pmatrix} \geq 0; A.\begin{pmatrix} I \\ N \\ 1 \end{pmatrix} = X\}$$

Describing the accessed data reduces to look for the set of values of X for which the set of iterations I accessing X through A is not empty. In other words, we seek the domain of existence in $P(I, N)$ of an integer point in the inverse image (also called *preimage*) of X by A:

$$\{X \in \mathbb{Z}^m \mid \mathit{preimage}(X, A) \cap P(I, N) \neq \emptyset\}$$

First, there must exist integer points in $\mathit{preimage}(X, A)$. Then, we must see for which values of X these points are in $P(I, N)$.

Existence of an integer point in the preimage

For simplicity, assume there is no equality in the definition of P (we have seen in section 2.5.3 how to come to such a case). We look for integer values of N and X such that there exists an integer solution I to

$$A.\begin{pmatrix} I \\ N \\ 1 \end{pmatrix} = X \Leftrightarrow A_I I + (A_N N - \mathbb{I}_m X) + A_1 = 0 \tag{3.1}$$

where A_I (respectively A_N and A_1) is the part of A corresponding to I (respectively N and the constants), and $\mathbb{I}_m$ is the m-dimensional identity matrix. This problem is solved in section 2.5.3, where the $n - m$ equations are used to eliminate $n - m$ variables. Here, we are going to the same projection process, pointing out different aspects of the computation at each step. Let I_2 the variables to be eliminated and I_1 the variables to remain. Considering (N, X) as parameters, the values of (N, X, I_1) for which there exists an integer I_2 define an affine compression on (N, X, I_1):

$$\begin{pmatrix} N \\ X \\ I_1 \end{pmatrix} = G.\begin{pmatrix} N' \\ X' \\ I_1' \end{pmatrix} + \begin{pmatrix} N_0' \\ X_0' \\ I_0' \end{pmatrix}.$$

As G is lower triangular, X' depends on X and N, whereas N' depends only on N. This compression can then be read as separate conditions for the existence of an integer solution I_2 to (3.1):

- a constant condition on N,
- a parametric condition on X,
- a condition on I_1 depending on X and N.

The condition on N gives the values of the parameters for which a data can be accessed. The condition on X defines the lattice of array elements that can be accessed through the reference A. The condition on I_1 will allow to project P along I_2 so that there is a one-to-one mapping between the integer solutions $I = (I_1, I_2)$ to (3.1) and the integer points I_1' of the projected polyhedron.

Example 3.6. The polyhedron presented in example 2 has only one equation: $\{6i + 9j - 7 = k\}$. Let j the variable to be eliminated. The values of (i, k) for which there is an integer solution j to this equation are defined by the following compression:

$$\begin{cases} p \in \mathbb{Z} \\ k = 3k' + 2 \\ i = 3i' + 2k' \end{cases}, i', k' \in \mathbb{Z}.$$

The accessed array elements k defined by this compression are the same as in [73] and [17]. We can make this change of variables from (i, k) to $(i'k')$ explicit, giving: $2i' + j + k' - 1 = 0$.

□

The change of variables from X to X' defines the lattice of data that can be accesssed by an iteration through the reference A. It corresponds to a compression of the data space: in the new data space, each array element X' can be accessed by an integer iteration point I through A. But this I does not necessarily belong to the iteration domain $\{P(I, N) \geq 0\}$. In next subsection, we look for the array elements X' actually accessed by the loop nest.

Exact data accessed by the loop nest

In the compressed space (I_1', I_2, X', N'), there exist integer points in the preimage of any integer X' by function A for any integer value of N'. We are only interested in the integer points $I \in P$ whose image by A is integer: $E(I'_1, I_2, X', N') = P(I'_1, I_2, X', N') \cap \{A(I'_1, I_2, X', N') = 0\}$. More precisely, we want to know if there exist such points for given integer X' and N'. According to section 2.5.3, we can eliminate I_2 in E by using its $n-m$ equalities. Then, these equalities give the corresponding value of (I'_1, I'_2) for any value of I'_1. The resulting polyhedron $E(I'_1, X', N')$ is n-dimensional.

Example 3.7. In the intersection between the compressed polyhedron and the access equalities:

$$\begin{cases} 1 - 2k' \leq 3i' \leq 8 - 2k' \\ 1 \leq j \leq p \\ 2i' + j + k' - 1 = 0 \end{cases},$$

the equation can be used to eliminate j, giving:

$$P_3 = \begin{cases} 1 - 2k' \leq 3i' \leq 8 - 2k' \\ -k' - p + 1 \leq 2i' \leq -k' \end{cases}$$

□

Obviously, there exists an integer point I'_1 in a parametric polyhedron $E(I'_1, X', N')$ if and only if the (parametric) convex hull of its integer points, i.e., its integer hull, is not empty. It is shown in chapter 2 that the (parametric) integer hull of a polyhedron is the convex hull of the extremal integer points derived from its pseudo-vertices. The integer hull is non-empty if and only if there exist such extremal points. So a straightforward way to obtain the values of X' for which there exists an integer point I'_1 in $E(I'_1, X', N')$ would be to compute the values of X' for which $E(I'_1, X', N')$ has pseudo-vertices. For a given pseudo-vertex, these values are given by the validity domain of the pseudo-vertex (a.k.a. its *projected pseudo-vertex*).

Example 3.8. The pseudo-facet w.r.t. i' of P_3 derived from the inequality $2i'+k'+p-1 \geq 0$ is:

$$P_{3,1} = \begin{cases} 2i' + k' + p - 1 - (k' + p - 1) \ mod \ 2 = 0 \\ \begin{cases} k' - 3p + 1 + 3(k' + p - 1) \ mod \ 2 \geq 0 \\ -k' + 3p + 13 - 3(k' + p - 1) \ mod \ 2 \geq 0 \\ p - 1 - (k' + p - 1) \ mod \ 2 \geq 0 \end{cases} \end{cases}$$

$$= \begin{cases} 2i' + k' + p - \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}_{p,k'} = 0 \\ \begin{cases} k' - 3p + \begin{bmatrix} 4 & 3 \\ 3 & 4 \end{bmatrix}_{p,k'} \geq 0 \\ -k' + 3p + \begin{bmatrix} 10 & 13 \\ 13 & 10 \end{bmatrix}_{p,k'} \geq 0 \\ p - \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}_{p,k'} \geq 0 \end{cases} \end{cases}$$

As the value of all its variables are determined by equalities, it is a pseudo-vertex. The values of k' and p for which this pseudo-vertex belongs to P_3 (i.e., its validity domain) are given by the inequalities.

□

As there exists an integer point in E as soon as there exists a pseudo-vertex, the values of X' and N' such that there exists an integer point I'_1 in E can be given by the union of the validity domains of the pseudo-vertices of E.

Computing all the pseudo-vertices is not necessary. The existence of an integer point in E do not reduce to the existence of *all* its pseudo-vertices, but to the existence of *at least one* pseudo-vertex. Given a (hierarchically) linear order $\mathcal{O}$ on the variables I'_1, there exists an integer point in E if and only if there is always at least one integer point that is the maximum w.r.t. $\mathcal{O}$. Thus, the existence of an integer point in E reduces to the existence of a pseudo-vertex that is maximal w.r.t. $\mathcal{O}$.

Example 3.9. For P_3, we can for instance maximize $-i$, i.e. to minimize i. Two pseudo-vertices minimize i': $P_{3,1}$ and

$$P_{3,2} = \begin{cases} 3i' + 2k' - 1 - (2k' - 1) \; mod \; 3 = 0 \\ \begin{cases} 7 - (2k' - 1) \; mod \; 3 \geq 0 \\ k' - 2 - 2((2k' - 1) \; mod \; 3) \geq 0 \\ -k' + 3p - 1 + 2(2k' - 1) \; mod \; 3 \geq 0 \end{cases} \end{cases} = \begin{cases} 3i' + 2k' - \begin{bmatrix} 3 & 2 & 1 \end{bmatrix}_{k'} = 0 \\ \begin{cases} \begin{bmatrix} 5 & 6 & 7 \end{bmatrix}_{k'} \geq 0 \\ k' - \begin{bmatrix} 6 & 4 & 2 \end{bmatrix}_{k'} \geq 0 \\ -k' + 3p + \begin{bmatrix} 3 & 1 & -1 \end{bmatrix}_{k'} \geq 0 \end{cases} \end{cases}$$

The values of k and p such that there is an integer point in P_3 are then given by the union of the validity domains of $P_{3,1}$ and $P_{3,2}$:

$$P'_{3,1} = \begin{cases} k' - 3p + \begin{bmatrix} 4 & 3 \\ 3 & 4 \end{bmatrix}_{p,k'} \geq 0 \\ -k' + 3p + \begin{bmatrix} 10 & 13 \\ 13 & 10 \end{bmatrix}_{p,k'} \geq 0 \\ p - \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}_{p,k'} \geq 0 \end{cases} \cup P'_{3,2} = \begin{cases} k' - \begin{bmatrix} 6 & 4 & 2 \end{bmatrix}_{k'} \geq 0 \\ -k' + 3p + \begin{bmatrix} 3 & 1 & -1 \end{bmatrix}_{k'} \geq 0 \end{cases}$$

The fact that the union of $P'_{3,1}$ and $P'_{3,2}$ describe the exact set of integer values of k' such that there exists an integer solution in P_3 is not obvious. To convince the reader, let us show how, for instance, we can compute the data holes. The periodic numbers used in the definition of these validity domains are a short notation. These periodic polyhedra can be written more explicitly as:

$$P'_{3,1} = \begin{cases} \{3p - 4 \leq k' \leq 3p + 10; p \geq 2\} \; if \; k' \; mod \; 2 = 0 \; and \; p \; mod \; 2 = 0 \\ \{3p - 3 \leq k' \leq 3p + 13; p \geq 1\} \; if \; k' \; mod \; 2 = 1 \; and \; p \; mod \; 2 = 0 \\ \{3p - 3 \leq k' \leq 3p + 13; p \geq 1\} \; if \; k' \; mod \; 2 = 0 \; and \; p \; mod \; 2 = 1 \\ \{3p - 4 \leq k' \leq 3p + 10; p \geq 2\} \; if \; k' \; mod \; 2 = 1 \; and \; p \; mod \; 2 = 1 \end{cases}$$

$$P'_{3,2} = \begin{cases} \{6 \leq k' \leq 3p + 3\} \; if \; k' \; mod \; 3 = 0 \\ \{4 \leq k' \leq 3p + 1\} \; if \; k' \; mod \; 3 = 1 \\ \{2 \leq k' \leq 3p - 1\} \; if \; k' \; mod \; 3 = 2 \end{cases}$$

Each element of this decomposition is the intersection of a polyhedron and an integer lattice (defined, for instance, by $k' \; mod \; 2 = 0$ and $p \; mod \; 2 = 0$), *i.e.*, a $\mathbb{Z}$-polyhedron. So the data holes can be found in the following way: let $P_{k'}$ be the projection of P_3 on the space of k': $P_{k'} = \{2 \leq k' \leq 3p + 13\}$. The data holes can be determined for each validity domain by scanning the different lattices defined by the periodic numbers. For a given lattice L, the holes generated by the $\mathbb{Z}$-polyhedron P'_L whose lattice is L is given by $(P_{k'} \cap L) \setminus P'_L = (P_{k'} \setminus R) \cap L$. Here, the holes generated by $\{3p - 4 \leq k' \leq 3p + 10; p \geq 2, k' \; mod \; 2 = 0, p \; mod \; 2 = 0\}$ are: $P_{k'} \setminus \{3p - 4 \leq k' \leq 3p + 10; p \geq 2\} \cap \{k' \; mod \; 2 = 0, p \; mod \; 2 = 0\} = \{3p+11 \leq k' \leq 3p+13\} \cap \{k' \; mod \; 2 = 0, p \; mod \; 2 = 0\} = \{k' = 3p+12\}$, which is one of the data holes we were looking for ($k = 3k' + 2 = 9p + 38$).

□

Identifying the data holes determine the data computed by a loop nest more precisely that just by projecting along the access function the polyhedron defined by the loop bounds. In the example developed here, if the only array elements that are used after the loop nest execution are the elements $a[11]$ and $a[3p+12]$, the whole loop nest is useless and can be eliminated. Analyzing this can be considered costly when considering a single loop nest whose volume of accessed data is large. In this case, the ratio between the number of data accessed and the number of holes shall tend to zero, as well as the chances for having to eliminate a whole loop nest. But, for instance, dead code elimination based on the analysis of liveness of array elements have to propagate along the whole analyzed program. The error associated to an approximate analysis propagates as well, and the final error may disallow a powerful dead code elimination.

The aim of this example of use of the pseudo-facets for determining holes is just to show the relevance of pseudo-facets in the polytope model: we did not try to devise a specific way to compute holes that would be optimal in performance.

Among many applications (several of them are shown in [73] and [16]), this allows for instance to determine the exact number of data accessed by a loop nest through a non-invertible array access function. Next section shows how the number of integer points in such a projection can be derived from the domain of existence of an integer point in a polyhedron.

3.6.2 Number of integer points in the projection of a $\mathbb{Z}$-polyhedron

Counting the number of integer points in the image of a $\mathbb{Z}$-polyhedron P by a non-singular integer affine transformation T is trivial, as it equals the number of integer points in P. We have seen that when the transformation is singular, i.e., it can be represented by a non-invertible matrix A, some *holes* appear in the image. After some adequate transformations, we have obtained the definition of a set of points X' such that there exists an integer point I in the projection of a polyhedron $P(I, N)$, depending on parameters N. These integer points X' are the image of the integer points of P. So counting the number of existing X' gives the number of points in the projection of the $\mathbb{Z}$-polyhedron P along A. Ehrhart polynomials [16, 26] give the number of integer points in a rational polyhedron. Thus, the operation is almost straightforward, the only difficulty coming from the fact that the set of points X' are defined by a periodic polyhedron.

Example 3.10. The set of integer points in E, that is to say in P_3, is defined in section 3.6.1 by $P'_{3,1} \cup P'_{3,2}$. $P'_{3,1}$ is periodic in function of k', with a period of 2, and $P'_{3,2}$ is also periodic along k' but with a period of 3. The couple $(P'_{3,1}, P'_{3,2})$ is then also periodic along k' with a period of 6. Similarly, it is periodic along p with a period of 2. We can then write $P'_{3,1} \cup P'_{3,2}$ as:

$$\begin{cases} 3p - \begin{bmatrix} 4 & 3 & 4 & 3 & 4 & 3 \\ 3 & 4 & 3 & 4 & 3 & 4 \end{bmatrix}_{p,k'} \leq k' \leq 3p + \begin{bmatrix} 10 & 13 & 10 & 13 & 10 & 13 \\ 13 & 10 & 13 & 10 & 13 & 10 \end{bmatrix}_{p,k'} \\ p \geq \begin{bmatrix} 2 & 1 & 2 & 1 & 2 & 1 \\ 1 & 2 & 1 & 2 & 1 & 2 \end{bmatrix}_{p,k'} \end{cases}$$

$$\cup \left\{ \begin{bmatrix} 6 & 4 & 2 & 6 & 4 & 2 \\ 6 & 4 & 2 & 6 & 4 & 2 \end{bmatrix}_{p,k'} \leq k' \leq 3p + \begin{bmatrix} 3 & 1 & -1 & 3 & 1 & -1 \\ 3 & 1 & -1 & 3 & 1 & -1 \end{bmatrix}_{p,k'} \right.$$

This system defines twelve $\mathbb{Z}$-polyhedra, each of them having its own Ehrhart polynomial. We can transform the system by considering that k' can always be written $k' = 6k'' + k' \ mod\ 6 = k'' + \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \end{bmatrix}_{k'}$, and similarly $p = 2p' + p \ mod\ 2 = 2p' + \begin{bmatrix} 0 & 1 \end{bmatrix}_p$:

$$\begin{cases} 6p' - \begin{bmatrix} 4 & 2 & 2 & 0 & 0 & -2 \\ 0 & 0 & -2 & -2 & -4 & -4 \end{bmatrix}_{p,k'} \leq 6k'' \leq 6p' + \begin{bmatrix} 10 & 12 & 8 & 10 & 6 & 8 \\ 16 & 12 & 14 & 10 & 12 & 8 \end{bmatrix}_{p,k'} \\ 2p' \geq \begin{bmatrix} 2 & 1 & 2 & 1 & 2 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}_{p,k'} \end{cases}$$

$$\cup \left\{ \begin{bmatrix} 6 & 3 & 0 & 3 & 0 & -3 \end{bmatrix}_{p,k'} \leq 6k'' \leq 6p' + \begin{bmatrix} 3 & 0 & -3 & 0 & -3 & -6 \\ 6 & 3 & 0 & 3 & 0 & -3 \end{bmatrix}_{p,k'} \right.$$

In this form, the variables over which the polyhedron is enumerated, namely k'' and p', are independent from $k' \ mod\ 6$ and $p \ mod\ 2$. Then, the number of distinct values of k' in $P'_{3,1} \cup P'_{3,2}$ is the sum of the six Ehrhart polynomials given by the distinct combinations of $k' \ mod\ 6$, for each value of $p \ mod\ 2$. Let us compute the Ehrhart Polynomial $\epsilon(p)$ for $p \ mod\ 2 = 0$:

$$\begin{cases} \{6p'-4 \le 6k'' \le 6p'+10; 2p' \ge 2\} \cup \{6 \le 6k'' \le 6p'+3\} & , k' \ mod\ 6 = 0 \\ \{6p'-2 \le 6k'' \le 6p'+12; 2p' \ge 1\} \cup \{3 \le 6k'' \le 6p'\} & , k' \ mod\ 6 = 1 \\ \{6p'-2 \le 6k'' \le 6p'+8; 2p' \ge 2\} \cup \{0 \le 6k'' \le 6p'-3\} & , k' \ mod\ 6 = 2 \\ \{6p' \le 6k'' \le 6p'+10; 2p' \ge 1\} \cup \{3 \le 6k'' \le 6p'\} & , k' \ mod\ 6 = 3 \\ \{6p' \le 6k'' \le 6p'+6; 2p' \ge 2\} \cup \{0 \le 6k'' \le 6p'-3\} & , k' \ mod\ 6 = 4 \\ \{6p'+2 \le 6k'' \le 6p'+8; 2p' \ge 1\} \cup \{-3 \le 6k'' \le 6p'-6\} & , k' \ mod\ 6 = 5 \end{cases}$$

$$= \begin{cases} \{2 \le 6k'' \le 6p'+10; p' = 1\} \cup \{6 \le 6k'' \le 6p'+10; p' \ge 2\} & , k' \ mod\ 6 = 0 \\ \{3 \le 6k'' \le 6p'+12; 2p' \ge 1\} & , k' \ mod\ 6 = 1 \\ \{0 \le 6k'' \le 6p'+8; p' \ge 1\} & , k' \ mod\ 6 = 2 \\ \{3 \le 6k'' \le 6p'+10; 2p' \ge 1\} & , k' \ mod\ 6 = 3 \\ \{0 \le 6k'' \le 6p'+6; p' \ge 1\} & , k' \ mod\ 6 = 4 \\ \{-3 \le 6k'' \le 6p'-6; 2p' \ge 1\} \cup \{6p'+2 \le 6k'' \le 6p'+8; 2p' \ge 1\} & , k' \ mod\ 6 = 5 \end{cases}$$

Their number of integer points are then:

$$\begin{cases} 2\ if\ p' = 1, p'+2\ if\ p' \ge 2 & , k' \ mod\ 6 = 0; & p'+2\ if\ p' \ge 1 & , k' \ mod\ 6 = 1 \\ p'+2\ if\ p' \ge 1 & , k' \ mod\ 6 = 2; & p'+1\ if\ p' \ge 1 & , k' \ mod\ 6 = 3 \\ p'+2\ if\ p' \ge 1 & , k' \ mod\ 6 = 4; & p'+1\ if\ p' \ge 1 & , k' \ mod\ 6 = 5 \end{cases}$$

Summing all these possible values over $k'\ mod\ 6$ gives the number of integer points in $P'_{3,1} \cup P'_{3,2}$ when $p\ mod\ 2 = 0$: $\epsilon(p) = \begin{cases} 16\ if\ p' = 1 \\ 6p'+10\ if\ p' \ge 2 \end{cases} = \begin{cases} 16\ if\ p = 2 \\ 3p+10\ if\ p \ge 4 \end{cases}$

3.7 Remarks and perspectives

Using the pseudo-facet machinery, we have defined a new way to compute the integer points that are maximum w.r.t. a hierarchically linear order in a parametric polyhedron. This frame encompasses the known problems of parametric integer linear programming and of the integer lexicographic maximum. In [68], Minoux stresses Gomory's observation: the number of integer points to be considered for finding an integer extremum w.r.t. $v.I$ for the problem $\{AI = b, I \ge 0\}$ is finite. Once known the rational extremum u as well as an optimal basis B, the number of integer points near u that can be an integer solution to the problem is bounded by $det(B)$. This observation corroborates ours, for which the number of integer points which can be the solution corresponding to a given rational point equals $\Pi_{k=1}^{n} p_k$, where p_k is the period along the variable i_k.

The algorithm works by a series of inequality selections, pseudo-facet computations and projections. The pseudo-facet computation and the projection steps are linked, as a pseudo-facet w.r.t. a variable i_k must be projected along i_k. The choice of the variable to be projected has an influence on the periodicity of the resulting pseudo-facet. This choice could be extended by developing the concept of pseudo-facet *w.r.t. a vector of* $\mathbb{Z}^n$. This idea is somehow close to the concept of *first/last faces* introduced by Clauss in [14] for synthesizing space-optimal systolic arrays. If this extension is possible, it raises the question of the best set of vectors for which the periodicity of the solution is minimal. This question will be considered in future works.

Besides, we believe that there are still some optimizations to be found in the presented algorithms. Better selection algorithms, and some ways to give a symbolic maximum have to be found in order to further reduce the algorithm's complexity.

Chapter 4

Further work on Ehrhart polynomials

4.1 Motivation

Ehrhart polynomials allow to enumerate the number of integer points in a parametric rational polyhedron. In the polytope model, loop nests are represented by unions of parametric rational polyhedra. Iterations of the loop nest correspond to integer points in a polyhedron P: counting the number of iterations executed by the loop nest boils down to computing the Ehrhart polynomial of P. This principle has been introduced by Clauss [15, 16], extended by Clauss and Loechner [19], and has found many applications, especially in loop nest parallelization and optimization.

- in [19], Clauss and Loechner extract the number of distinct processors used to execute a parametric nested loop with a given schedule. This number is an Ehrhart polynomial depending on the loop parameters. Together with Wilde, they also show in [21] that Ehrhart polynomials is the most appropriate technique to compute the exact parametric number of integer points in a polyhedron, by solving problems that were addressed by other techniques until then [35, 82, 73, 80].
- Boulet and Redon [9] use Ehrhart polynomials to evaluate the number of data communicated during the execution of a parallelized loop nest in HPF according to a given array alignment, depending on the chosen data distribution. As this evaluation

is done without having to execute the program, the program writer can then choose an optimal data distribution for his/her HPF program.

- Loechner and Mongenet [59] compute the number of communications resulting from a space-time mapping of a loop nest, and use the resulting Ehrhart polynomial as a criterion for choosing the best space-time mapping.

- Bourgeois, Spies and Tréhel [10] use Ehrhart polynomials in the static part of their performance predictor for C/MPI programs *ChronosMix*.

- We showed in [22] a method for ranking the iterations of a loop nest according to their execution order. They use this ranking, defined by an Ehrhart polynomial, to organize array elements that are accessed by a loop nest in the exact order in which they are accessed. The consequence is an increase of the data locality of memory accesses. We extended the method from the case of one loop nest accessing arrays through an invertible array reference (so with no self-reuse) to the case of multiple loops accessing array elements through multiple non-invertible array references [20, 57, 58], i.e., cases where self-reuse is the dominating temporal locality factor. We treated separately the cases when only group-reuse can be exploited [67, 64, 65].

- Braberman, Garbervetsky and Yovine [11] use Ehrhart polynomials to evaluate the amount of dynamic memory needed by Java methods, in order to allow a precise scoped memory management and to allocate objects only when they are needed.

- Seghir et al [77] use Ehrhart polynomials to synthesize cache hints at compile time, by computing the number of distinct array elements accessed between two distinct accesses to an array element.

- Verdoolaege et al [83] focus on the use Ehrhart polynomials in the context of embedded systems.

But a series of techniques would only need a good approximation of the number of points in a polyhedron.

- Ju, Collard and Oukbir [44] determine the probability for an array data dependence to occur in a window of some successive iterations, in order to decide whether a speculative load (prefetch) is profitable or not. Roughly, this probability depend on a ratio between the number of iterations for which the dependence occurs and the total number of iterations in the loop nest. They do not use Ehrhart polynomials in their implementation, but refer to them as an appropriate method. The authors look for *quick approximations* of the probabilities: the accuracy of Ehrhart polynomials is then not really needed there.

- Pingali et al [48] mention the use of Ehrhart polynomials for a tool to estimate the number of cache lines accessed by some array references in loop nests. This number is needed to derive the size of data blocks that will be grouped to increase the locality of their accesses and then increase the performance of the program. But their implementation use bounding boxes for the considered polyhedra, which give a very rough over-approximation of its Ehrhart polynomial.

- Hannig and Teich [39] approximate the number of used processors corresponding to a given parallel space-time mapping with Ehrhart polynomials: the authors assume that the projection of integer points of a polytope is given by the integer points that belong to the projection of the polyhedron. The Ehrhart polynomial computed this way is an over-approximation of the actual number of integer points on which the iterations are mapped.

- The same authors [40] use Ehrhart polynomials *approximation* to estimate the energy consumption of VLSI components in embedded systems.

- Lisper [54] also uses Ehrhart polynomials to evaluate the worst case execution time of a program.

Other authors even compute approximations of the number of integer points in a polyhedron.

- Tawbi developed a method for enumerating the integer points of a polyhedron, by expressing it as sums which are then decomposed using Bernoulli formulae. The result is an approximation in the case of rational constraints. This technique has been used to tackle the same kind of problems as with Ehrhart polynomials, for instance for estimating the time of execution of a loop nest [81, 82].

- Van Engelen, Gallivan and Walsh [27] estimate the worst case execution time of a loop nest by finding a polynomial interpolation of the number of iterations. In the case of polyhedra with rational vertices (or equivalently when non-unit loop strides are considered), this polynomial is an approximation of the number of integer points in the polyhedron. This method also tries to handle polynomial loop bounds, in which case it generally leads to an approximation of the number of iterations.

- Heine and Slovik [42] use Ehrhart polynomials to determine the best loop transformation and data to be distributed to a cluster of workstations in a *cyclic block* manner (as in HPF). They first compute a set of candidate mappings of iterations to processors according to individual data access functions. Then they compute the difference between local data and remote data to be accessed by the processors, according to each mapping. This difference, which is an Ehrhart polynomial, is used to rank the candidate mappings symbolically and then to choose the optimal mapping. The authors simplify the Ehrhart polynomials by replacing each periodic coefficient by its average value. The relevance of this approximation method is discussed in section 4.8

Unfortunately, no formal upper bound is given on the approximation error for these different works, to our knowledge.

Some exact approaches also exist.

- Pugh's [73] algorithm combines Tawbi's approach with the omega test for simplifying the input, which is a set of Presburger formulas. Pugh handles rational bounds by treating integer parts as symbolic, transforming them into modulo functions.

- Sakellariou [75] goes further into decomposing nested sums, following the work of Tawbi. His goal was to split loop nests into balanced sub-loops for parallelizing the corresponding computations. While being exact in cases where Tawbi had to approximate the result, his framework cannot deal with a polyhedron as is, but only when stated as bounds of loop nests. Hence, a Fourier-Motzkin preprocess is needed. As the bounds resulting from such a preprocess are often expressed as a maximum or a minimum among affine loop bounds, the proposed algorithm must split the polyhedron to have only affine loop bounds. The result is less simple and clear than Ehrhart polynomials.

Until recently, the only implemented algorithm for computing Ehrhart polynomials of a parametric rational polyhedron was using the Clauss-Loechner-Wilde interpolation method, implemented in the polyhedral library *Polylib* [84]. Barvinok [5] proposed a non-parametric decomposition of cones into simplicial unimodular cones, that allows to compute the Ehrhart polynomial of a non-parametric polytope in polynomial time for a fixed dimension. The resulting algorithm is the last step of a series of advances in enumerative combinatorics, to which belong the works of Brion, Dyer, Kantor, Khovanskii, Lawrence, Pommersheim and Pukhlikov, which are summed up in [6]. It has been optimized and implemented in the lattE software [61]. Seghir and Verdoolaege [77, 83] found simultaneously the same idea to extend this algorithm to the parametric case. Firstly using the explicit form of the periodic numbers, the computation time of this parametric extension is still considered of a too high complexity, while being better than the Clauss-Loechner-Wilde method on the presented examples. But when keeping them under a symbolic form (with modulos, similarly to Pugh and at a smaller degre to us), the computation complexity decreases significantly.

The need for an approximation is not only motivated by computation complexity reasons, but also by the need for a result having a simpler mathematical structure. Also, many mathematical tools that can analyze and manipulate polynomials exist. Ideally, we would like to compute a *good* approximation of the Ehrhart polynomial of P in polynomial time. The quality of the approximation must be known: the user should have a clear bound on the approximation error, which should ideally be asymptotically zero when possible. This chapter studies some aspects of Ehrhart polynomials that can lead to an approximation by two distinct ways. Other side results are also shown: an optimization for the Clauss-Loechner-Wilde interpolation method and an alternative Ehrhart polynomial computation method for particular polyhedra.

First, in section 4.2 we have a look on a (proved) conjecture stated by Ehrhart that links the periodic character of the Ehrhart polynomial of a polyhedron P to some geometric properties of its faces. We show an extension of this conjecture in section 4.3 that gives a less restrictive condition for a polyhedron to have a non-periodic Ehrhart polynomial $\mathcal{E}$. Then, we decompose the problem of giving an approximation of $\mathcal{E}$ into two subproblems: finding the values of the parameters for which the affine hull of P contains an integer point (section 4.4), and then transforming P so that it has a non-periodic Ehrhart polynomial, which allows to derive a non-periodic approximation. Two kinds of affine transformations are considered: a compression of the parameters space of P in section 4.5, and an expansion of the variables space of P in section 4.6. A fast tailor-made algorithm to compute the Ehrhart polynomial of a polyhedron resulting from this variable space expansion is presented in section 4.7. They are compared to other existing and possible approximation methods in section 4.8. Finally, section 4.9 sums up the contribution brought by this chapter, and gives future working directions.

4.2 A focus on Ehrhart's conjecture

The approximation method presented here is based on the notion, due to Ehrhart, of the *grade* of an Ehrhart polynomial w.r.t. a parameter a: it is the highest degree monomial having a periodic coefficient in function of a.

Example 4.1. The grade w.r.t. a of the following Ehrhart Polynomial :

$$12a^3 - 5a^2 + \begin{bmatrix} 2 & 3 \end{bmatrix}_a a + 1$$

is 1, as all the coefficients of degree strictly higher than 1 are non-periodic.

Ehrhart has conjectured [26] the following relationship between the grade of an Ehrhart polynomial and a geometric property of the polyhedron, which has been proved later by Stanley [79] and McMullen [63]:

Theorem 4.1 (Ehrhart's conjecture). *Let $\mathcal{E}(a)$ be the Ehrhart polynomial of a polytope P, in a n-dimensional space, depending on parameter a:*

$$\mathcal{E}(a) = c_k(a)a^k + c_{k-1}(a)a^{k-1} + \ldots + c_0(a),$$

where $c_q(a), q \in [0..k]$ are periodic numbers depending on a. If, for $q \in [0..k]$, the affine hulls of the q-dimensional faces of P contain an integer point for any value of the parameter a, then the grade of $\mathcal{E}(a)$ equals $q - 1$.

A corollary of this theorem is: if the affine hulls of all the 0-dimensional faces of a polytope P are integer for any values of the parameters, the grade of its Ehrhart polynomial is -1. That is to say, if the vertices of P are integer for any values of the parameter, then the Ehrhart polynomial is non-periodic. Another corollary is that if the affine hull of P is $\mathbb{Q}^n$ (i.e., P is full-dimensional), then the term of $\mathcal{E}(a)$ of degree n is non-periodic.

This theorem extends to a polytope with any number of parameters, by applying it parameter by parameter. The Ehrhart polynomial is then multivariate, so it is more convenient to use a vectorial representation. Parameters will be noted $N = (n_k) \in \mathbb{Z}^p$. We use an index vector $r = (r_k) \in \mathbb{Z}^p$ to distinguish monomials. Any term $n_1^{r_1} n_2^{r_2} \dots n_p^{r_p}$ written N^r, r is called the multi-index of N. Let d_k be the maximum degree along n_k in the polynomial, and let $d = (d_k) \in \mathbb{Z}^p$. A multivariate Ehrhart polynomial can be written as:

$$\mathcal{E}(N) = \sum_{0 \leq r \leq d} \alpha_r N^r,$$

where the $\leq$ operator is element-wise, and α_r is a (multidimensional) periodic number. The grade g_k of $\mathcal{E}(N)$ along n_k is then the highest degree r_k of the monomials whose coefficients are periodic along n_k. The vector $g = (g_k) \in \mathbb{Z}^p$ is then called the *(multi-)grade* of $\mathcal{E}(N)$. Then, considering multiple parameters, theorem 4.1 becomes:

Theorem 4.2 (p-dimensional Ehrhart's conjecture). *Let $\mathcal{E}(n)$ be the Ehrhart polynomial of a polytope P depending on parameters $N \in \mathbb{Z}^p$. If there exists some value $N_0 = (n_{1,0}, n_{2,0}, \dots, n_{k,0}, \dots, n_{p,0}), k \in [1..p]$ of the parameters such that the affine hulls of all the q_k-dimensional faces ($q_k \in [1..d_k]$) of P contain an integer point for any $N = (n_{1,0}, n_{2,0}, \dots, n_k, \dots, n_{p,0}), n_k \in \mathbb{Z}$, then the grade $g = (g_k) \in \mathbb{Z}^p$ of $\mathcal{E}(N)$ is bounded by $g_k < q_k$.*

This theorem is more general but also more intricate than theorem 4.1. The following corollaries may be of simpler usage, while being less precise:

Corollary 1. *If, for $q \in [0..n]$, the affine hulls of all the q-dimensional faces of P contain an integer point for any values of N, then $g \leq (q, q, \cdots, q)$ (element-wise).*

As we are interested in non-periodic Ehrhart polynomials, a condition for an Ehrhart polynomial to be non-periodic can be straightforwardly derived:

Corollary 2. *If all the vertices of P are integer for any values of N, then the Ehrhart polynomial of P is non-periodic.*

This last corollary is similar to the first corollary we have derived from the one-dimensional theorem 4.1.

4.3 A more precise condition for an Ehrhart polynomial to be non-periodic

In chapter 2, we presented a method for computing the integer hull P' of a parametric rational polyhedron P, i.e., the convex hull of its integer points. It is shown that P' is a periodic polyhedron: it is periodically defined by one of the polyhedra of a finite set $\{P_1, P_2, \ldots, P_t\}, t \in \mathbb{N}^+$ of polyhedra with integer vertices. As each of these polyhedra have integer vertices (for any values of the parameters), their Ehrhart polynomial is non-periodic. Hence, the periodicity of the Ehrhart polynomial of P is the periodicity of the integer hull of P.

Let P be defined by a non-redundant set of constraints (equalities and inequalities):

$$A_P I + B_P N + C_P \underset{\geq}{=} 0 \tag{4.1}$$

By construction, the periodicity of P' is independent from the constant part C_P of P. It follows that the periodicity of the Ehrhart polynomial of P does neither depend on the constant part of P. This observation was also showed by Ehrhart [26] who revealed that the number of integer points can be defined by a linear recurrence relation on the parameters over a validity domain. This allows to extend theorem 4.1.

The affine hull of a q-dimensional face f_q of a parametric rational polyhedron P of variables $I \in \mathbb{Z}^n$ and of parameters $N \in \mathbb{Z}^p$ is given by turning a set of $m = n - q$ constraints of (4.1) into equalities:

$$AI + BN + C = 0,$$

where A, B, and C are respectively $m \times n, m \times p$, and $m \times 1$ integer matrices, so that A is full-row-rank (if not, f_q would be empty).

Theorem 4.1 derives periodicity information for the Ehrhart polynomial of P from the fact that the affine hulls of q-dimensional faces of P contain an integer point for any values of N. Let $A = [H_A\ 0]U_A$ be the left Hermite Normal form of A. According to section 1.2, there exists an integer solution for I if and only if $H_A^{-1}.(BN + C) \in \mathbb{Z}^m$. Equivalently, there is an integer point in the affine hull of the face f_q if and only if $H_A^{-1}(BN) \in \mathbb{Z}^m$ and if $H_A^{-1}C \in \mathbb{Z}^m$.

Theorem 4.1 tells that if all the q-dimensional faces of P fulfill these two conditions - on B and on C - then the grade of the Ehrhart polynomial $\mathcal{E}(N)$ of P is strictly less than

q. But we have seen that the periodicity of $\mathcal{E}(N)$ is independent from the constant part C of P. Hence, condition $H_A^{-1}C \in \mathbb{Z}^m$ has no influence on the grade of P and then is not necessary in theorem 4.1. Moreover, we can consider the condition $H_A^{-1}(BN) \in \mathbb{Z}^m$ parameter by parameter, as the k^{th} column-vector $B_{\bullet k}$ of B corresponds to parameter n_k. It comes the following theorem:

Theorem 4.3 (extension of p-dimensional Ehrhart's conjecture). *Let $\mathcal{E}(N)$ be the Ehrhart polynomial of a rational polyhedron P parameterized by $N = (n_k) \in \mathbb{Z}^p$. If, for any q-dimensional face of P whose affine hull is defined by:*

$$AI + BN + C = 0,$$

where $I \in \mathbb{Z}^n$ are the variables, and A, B and C are respectively $(n-q)\times n, (n-q)\times p$ and $(n-q)\times 1$ integer matrices, we have:

$$H_A^{-1}B_{\bullet k}n_k \in \mathbb{Z}, \forall n_k \in \mathbb{Z},$$

where $A = [H_A\ 0]U_A$ denotes the left Hermite normal form of A and $B_{\bullet k}$ is the k^{th} column-vector of B, then the grade of $\mathcal{E}(N)$ along n_k is strictly less than q.

The corollary giving a condition for an Ehrhart polynomial to be non-periodic is then:

Corollary 3 (non-periodicity condition). *If all the vertices of P, defined by*

$$AI + BN + C = 0,$$

where $I \in \mathbb{Z}^n$ are the variables, and A, B and C are respectively $n\times n, n\times p$ and $n\times 1$ integer matrices, we have:

$$H_A^{-1}BN \in \mathbb{Z}^n, \forall N \in \mathbb{Z}^p,$$

where $A = [H_A\ 0]U_A$ denotes the left Hermite normal form of A, then $\mathcal{E}(N)$ is non-periodic.

Alternatively, this condition can be written

$$H_A = H_{AB},$$

where $(A\ B) = H_{AB}U_{AB}$ is the left Hermite normal form of the matrix $(A \mid B)$. It can be seen as following : if any linear combination of the vectors of A and B can be reached by a linear combination of the vectors of A, then there is an integer solution to $AI + BN = 0$ for any values of N, and conversely.

We want to compute an approximation of the Ehrhart polynomial $\mathcal{E}_P(N)$ of a rational parametric polyhedron P, which does not, in general, satisfy the non-periodicity condition given by corollary 3. The principle we choose is to transform P into a polyhedron P' satisfying this condition, whose Ehrhart polynomial $\mathcal{E}_{P'}$ is then non-periodic, by transforming affinely either its parameter space (in section 4.5) or its variable space (in section 4.6). Then, the polynomial that approximates $\mathcal{E}_P(N)$ is obtained by coming back to the original space. But first, we focus on the particular case of a non-full-dimensional polyhedron, and show how this case can be reduced to the case of a full-dimensional polyhedron.

4.4 Non-full-dimensional polyhedra

Loechner and Wilde [55] have shown that the parameter space of a parametric rational polyhedron P can be partitioned into polyhedral domains (called *validity domains*) in which P can be expressed as a non-redundant set of equalities and inequalities over variables $I \in \mathbb{Q}^n$ and parameters $N \in \mathbb{Z}^p$. In section 2.5.2, we have seen that a set of inequalities can be equivalent to an equality for some values of the parameters. These couples of inequalities are called *implicit equalities*, as in [76], by opposition to plain (*explicit*) equalities. It is shown that the definition domain can be split into parameters domains with a certain set of implicit equalities, and how these inequalities can be transformed into equivalent explicit equalities. Here, implicit equalities are assumed to be converted into explicit equalities: the considered values of the parameters belong to one of such domains. Also, without loss of generality, we consider one fixed Loechner-Wilde validity domain.

The e equalities define the hyperplane on which the polyhedron lies, and the inequalities partition this hyperplane by half-spaces. The *geometric dimension* of P is given by $n - e$: it is the dimension of its affine hull $aff(P)$, the hyperplane $\epsilon(N) \in \mathbb{Q}^n$ defined by the equalities of P:

$$A_e I + B_e N + C_e = 0, \tag{4.2}$$

P is called *full-dimensional* when its affine hull is $\mathbb{Q}^n$, i.e., when $e = 0$.

In the general case of a non-full-dimensional polyhedron, P contains integer points only for some values of N, defined by an integer lattice. Knowing this lattice allows:

- to get rid of the (easily large number of) values of N for which $aff(P)$ contains no integer point,

- to project P into a space in which it is full-dimensional, and for which any integer point corresponds to an integer point of the affine hull of P (as presented in section 2.5.3).

According to section 1.2, there is an integer point in $\epsilon(N)$ if and only if:

$$H_{A_e}^{-1}(BN + C) \in \mathbb{Z}^e, \tag{4.3}$$

where $A_e = H_{A_e}U$ is the Hermite normal form of A_e. According to section 2.5.3, is given by:

$$N = GN' + N_0, \tag{4.4}$$

where G is an invertible lower triangular integer matrix whose non-zero elements are positive.

If no N_0 exists, there is no value of the parameters yielding an integer solution for I. The considered polytope contains then no integer point for any value of the parameters.

Equation (4.4) defines:

- a condition on N for the existence of an integer solution to equation (4.2), i.e., for the existence of an integer point in P,

- a compression of the parameters space (from N to N') so that there exists an integer point in P for any value of N'.

In this section, we have seen that it is possible to compress the parameter space in order to have integer points in the affine hull of the resulting polyhedron for any values of the new parameters. Next section uses the same principle to enforce the non-periodicity condition by a linear parameter space compression.

4.5 Non-periodicity by parameter space compression

4.5.1 Non-periodicity for one vertex

In section 4.3, we have seen that $\mathcal{E}(N)$ is non-periodic if, for each vertex defined by $AI + BN + C \underset{\geq}{=} 0$, we have

$$H_A^{-1} BN \in \mathbb{Z}^n. \tag{4.5}$$

This problem is the same as in equation (4.3), solved in last section, but with a constant part equal to zero. The solutions in N to this problem are then defined by an integer lattice:

$$N = GN',$$

where G is a triangular nonnegative integer p-dimensional matrix. Notice that there always exists a solution, as the vector $(0, \cdots, 0)$ is always a particular solution.

4.5.2 Non-periodicity for the polyhedron

Let $\{v_1, v_2, \ldots, v_r\}$ be the r vertices of P, and let $N = G_i N'$ be the condition on N for a vertex $v_i, i \in [1..r]$ to satisfy the non-periodicity condition (4.5). The values of N where all the vertices satisfy (4.5) are given by the intersection of the r lattices for which $v_i, i \in [1..r]$ satisfy (4.5). It is given by the lattice

$$\begin{pmatrix} N \\ 1 \end{pmatrix} = G_v \cdot \begin{pmatrix} N' \\ 1 \end{pmatrix}, \tag{4.6}$$

where the column-vectors of matrix G_v span the intersection of the lattices spanned by the column-vectors of $G_i, i \in [1..r]$.

Equation (4.6) defines a change of variables from N to N', such that all the vertices of P satisfy equation (4.5) for each value of N'.

Example 4.2. The Ehrhart polynomial of the following polyhedron parameterized by n :

$$P_1(n) = \begin{cases} i \geq 0 \\ j \geq 0 \\ -2i + n \geq 0 \\ -2j + n + 1 \geq 0 \end{cases}$$

is given by

$$\mathcal{E}_{P_1}(n) = \frac{1}{4}n^2 + n + \begin{bmatrix} 1 & \frac{3}{4} \end{bmatrix}_n$$

P_1 has four vertices, for which we can identify matrices:

- $A_1 = \begin{pmatrix} -2 & 0 \\ 0 & -2 \end{pmatrix}, B_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, G_1 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$
- $A_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, B_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, G_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
- $A_3 = \begin{pmatrix} -2 & 0 \\ 0 & 1 \end{pmatrix}, B_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, G_3 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$
- $A_4 = \begin{pmatrix} 0 & -2 \\ 1 & 0 \end{pmatrix}, B_4 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, G_4 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$

The intersection of these lattices is then given by:

$$\begin{pmatrix} n \\ 1 \end{pmatrix} = G_v \begin{pmatrix} n' \\ 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} n' \\ 1 \end{pmatrix}$$

Changing n into n' in P_1 gives:

$$P_1'(n') = \begin{cases} i \geq 0 \\ j \geq 0 \\ -i + n' \geq 0 \\ -2j + 2n' + 1 \geq 0 \end{cases},$$

which satisfies the condition for its Ehrhart polynomial to be non-periodic. It is given by:

$$\mathcal{E}_{P_1'}(n') = n'^2 + 2n' + 1$$

Consider the vertex of P_1 of coordinates $\begin{pmatrix} n/2 \\ (n+1)/2 \end{pmatrix}$. Notice that there exists no linear compression on n such that this vertex can be integer. Hence, Ehrhart's conjecture (theorem 4.1) would not allow to find a sufficient condition on n for $\mathcal{E}_{P_1}(n)$ to be non-periodic. □

This change of variables defines a restriction of the values of N to a lattice on which the Ehrhart polynomial of P has a given (non-periodic) polynomial definition. Computing the Ehrhart polynomial of P on this lattice will then give a non-periodic integer-valued polynomial $\mathcal{E}'(N')$. Changing back the variables from N' to N will then give a rational-valued polynomial $\mathcal{E}'(N)$ whose values are integer when N satisfies condition (4.6). Let $\mathcal{E}_P(N)$ be the Ehrhart polynomial of P. As the compression defined by equation (4.6) is not affine but linear, we always have $N = 0 \Leftrightarrow N' = 0$, so the computed polynomial is the definition of $\mathcal{E}_P(N)$ for $N \ mod \ S = 0$, where S is the period of $\mathcal{E}_P(N)$. In order to avoid values of N for which $\mathcal{E}_P(N) = 0$ because the affine hull of P contains no integer points, we assume that the parameter space has beforehand been compressed using the technique presented in section 4.4.

Example 4.3. The change of variables to come back to n is given by:

$$\begin{pmatrix} n' \\ 1 \end{pmatrix} = G_v^{-1} \begin{pmatrix} n \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} n \\ 1 \end{pmatrix}.$$

The polynomial resulting from applying this change of variables to $\mathcal{E}_{P_1'}(n')$ is:

$$\mathcal{E}_{P_1'}(n) = \frac{1}{4}n^2 + n + 1.$$

It is integer-valued when $n = 2n'$ (n is even), for which values the non-periodicity condition is satisfied for P.

□

Next subsections present applications to this parameters compression technique and an interpretation of $\mathcal{E}_{P'}(N)$.

4.5.3 An optimisation for Ehrhart polynomials computation methods

G_v is the basis of a lattice of integer points N for which the number of points in P is given by the same non-periodic polynomial. Notice that this is true for any $N = N_1 + X, X \in \mathbb{Z}^p$, as the non-periodicity condition for $AI + BN_1 + (+BX + C) \underset{\geq}{=} 0$ is

$$N_1 = G_v N' \Leftrightarrow N = G_v N' + X,$$

which is then the general form of a lattice satisfying the non-periodicity condition for $AI + BN + C \underset{\geq}{=} 0$. The different values of X define $g = |det(G_v)|$ distinct lattices which partition $\mathbb{Z}^p$. The number of distinct non-periodic polynomials that define periodically $\mathcal{E}_P(N)$ is then g. Scanning the g distinct lattices would allow to compute Ehrhart polynomials, by computing these g polynomials separately and putting them together into $\mathcal{E}_P(N)$.

The periodicity $S = (s_k) \in \mathbb{Z}^p$ of $\mathcal{E}_P(N)$ is usually defined by:

$$\mathcal{E}_P(N) = \mathcal{E}_P(N + TX), X \in \mathbb{Z}^p,$$

where T is the p-dimensional integer diagonal matrix whose k^th diagonal element is s_k. G_v has the same property:

$$\mathcal{E}_P(N) = \mathcal{E}_P(N + G_vX), X \in \mathbb{Z}^p,$$

but it is lower triangular. One can derive T from G_v by an integer column elimination E:

$$T = G_vE.$$

As G_v already results from a unimodular column-elimination, E is unimodular only if G_v is diagonal. Else, it is non-unimodular of determinant e. In any case, we have:

$$t = det(T) = e * g \geq g,$$

which means that the periodicity information given by T is not minimal in the general case. The method proposed by Clauss for computing Ehrhart polynomials computes t polynomials and puts them together into $\mathcal{E}_P(N)$. The non-symbolic Seghir-Verdoolaege method also computes t polynomial values, but spares some computations by computing the periodic coefficients of the Ehrhart polynomial separately. The method suggested above for computing Ehrhart polynomials would then reduce the number of polynomials to be computed by a factor of e in the Clauss-Loecher-Wilde method, and a factor less than or equal to e in the non-symbolic Seghir-Verdoolaege method.

4.5.4 An approximation method for Ehrhart polynomials

We have seen that $\mathcal{E}_{P'}(N)$ is one of the polynomials defining $\mathcal{E}_P(N)$: the value of each of its coefficients is the value of the corresponding coefficient of $\mathcal{E}_P(N)$ for some values of

$N \; mod \; S$. Then, if $\mathcal{E}_P(N)$ has some non-periodic coefficients, the corresponding coefficients of $\mathcal{E}_{P'}(N)$ are exact for any values of N. If the affine hull of the d-dimensional polyhedron P contains integer points for any value of the parameters, then its term of degree d is non-periodic. According to section 4.4, we can always compress the parameters space so that the affine hull of P contains integer points for any value of the resulting parameters.

Then, $\mathcal{E}_{P'}(N)$ is an approximation of $\mathcal{E}_P(N)$ so that its terms of degree d are exact. Also, all the non-periodic coefficients of $\mathcal{E}_P(N)$ are exact in $\mathcal{E}_{P'}(N)$. However, unless the periodicity of the coefficients are computed, we do not know which of these coefficients will be exact or not.

This approximation method boils down to looking for the value of the Ehrhart polynomial for $N \; mod \; S = 0$, where S is the period for the parameters. The Clauss-Loechner-Wilde algorithm computes S (or a multiple of S) using the fact that the period along a given parameter n_k is given by the common denominator of the coefficients for n_k of the (rational parametric) coordinates of the vertices of P. Once S is known, compressing the parameters space of P according to S and computing the non-periodic Ehrhart polynomial of the resulting polyhedron gives the same approximation for $\mathcal{E}_P(N)$. As the Loechner-Wilde algorithm for computing the validity domains in which P has a given Ehrhart polynomial already computes the vertices of P, S can be quickly computed, as well as P'. Obviously, computing S as in Clauss-Loechner-Wilde's method and compressing P with S is the fastest method to compute P' and then $\mathcal{E}_{P'}(N)$.

4.6 Approximating by variable space transformations

4.6.1 Non-periodicity for one vertex

Given a vertex defined by the constraints

$$AI + BN + C \geq 0, \tag{4.7}$$

we would like to compute an affine transformation of the space of the variables $I \in \mathbb{Q}^n$ so that the non-periodicity condition,

$$H_A^{-1} BN \in \mathbb{Z}^n$$

is satisfied, which is the case if A is unimodular.

Let $A = UA'Q$ be the Smith normal form of A, where U and Q are unimodular matrices and A' is an integer diagonal matrix. Equation (4.7) rewrites:

$$UA'QI + BN + C \geq 0$$

The following change of variables :

$$A'QI = I', I' \in \mathbb{Z}^n$$

transforms equation (4.7) into :

$$UI' + BN + C \geq 0$$

As U is unimodular, we have $H_U^{-1}BN \in \mathbb{Z}^n$: there is an integer point I' in the new basis for any value of N.

Notice that the non-periodicity is conserved by further variable space expansion. Actually, making another change of variable $I' = X^{-1}I'', I'' \in \mathbb{Z}^n$, where X is an invertible integer matrix, transforms equation (4.7) into the following equalities:

$$UX^{-1}I'' + BN + C = 0$$

$$\Leftrightarrow I'' + XU^{-1}BN + XU^{-1}C = 0.$$

Then, as $XU^{-1}B$ is integer, the non-periodicity condition is still satisfied for any $I = Q^{-1}A'^{-1}X^{-1}I''$.

4.6.2 Non-periodicity for the polyhedron

But this expansion makes *one given vertex* of P satisfy the non-periodicity condition. From now on, we will note $L_q^{-1} = Q^{-1}A'^{-1}$ the expansion matrix for the q^{th} vertex, $q \in [1..r]$. Let $I = L_q^{-1}X_q^{-1}I''_q$ be the expansion of the variable space so that the non-periodicity condition is satisfied for the q^{th} vertex of P, $q \in [1..r]$. We look for an expansion of the variable space $I = L^{-1}I''$ that results from expansions of $I''_q, q \in [1..r]$. In other terms, the column-vectors of $L_q^{-1}, q \in [1..r]$ must be an integer linear combination of the column-vectors of L^{-1} : $L^{-1}X_q = L_q^{-1}$. In the lattice theory, the lattice spanned by the column-vectors of L^{-1} is called the *gcd* of the lattices spanned by L_q, and it is given by the left Hermite normal form of a matrix whose columns are the columns of all the L_q's, $q \in [1..r]$:

$$\begin{pmatrix} L_1^{-1} & L_2^{-1} & \cdots & L_r^{-1} \end{pmatrix} = \begin{pmatrix} L^{-1} & 0 \end{pmatrix} U_L,$$

where U_L is a unimodular integer matrix.

The Hermite normal form is usually computed for integer matrices, but it can be adapted to rational matrices by factor their terms by their common denominator. L^{-1} is the gcd of the q lattices, so it has the greatest possible determinant. Then, there exists no expansion derived from the expansions computed for the individual vertices that expands *less* than L^{-1}.

Example 4.4. The following polyhedron:

$$P_3 = \begin{cases} 2i - 3j + 2n - 4m - 1 \geq 0 & (a) \\ -i + 4j + 6m - 3 \geq 0 & (b) \\ -2i + 12n + 8m + 25 \geq 0 & (c) \end{cases},$$

whose variables are i and j and whose parameters are n and m, has three vertices, given by: $v_1 = (a) \cap (b), v_2 = (a) \cap (c), v_3 = (b) \cap (c)$. The Smith normal for Matrix $A_1 = \begin{pmatrix} 2 & -3 \\ -1 & 4 \end{pmatrix}$ (Matrix A for v_1) is :

$$A_1 = U_1 A_1' Q_1 = \begin{pmatrix} 2 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 5 \end{pmatrix} \begin{pmatrix} 1 & -4 \\ 0 & 1 \end{pmatrix},$$

giving the change of variables:

$$I_1' = A_1' Q_1' I = L_1 I = \begin{pmatrix} 1 & -4 \\ 0 & 5 \end{pmatrix} I \Leftrightarrow I = L_1^{-1} I_1' = \begin{pmatrix} 1 & \frac{4}{5} \\ 0 & \frac{1}{5} \end{pmatrix} I_1'.$$

Similarly, we have:

$$A_2 = \begin{pmatrix} 2 & -3 \\ -2 & 0 \end{pmatrix} = U_2 A_2' Q_2 = \begin{pmatrix} 1 & 0 \\ -4 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 6 \end{pmatrix} \begin{pmatrix} 2 & -3 \\ 1 & -2 \end{pmatrix} = U_2 L_2$$

$$I = L_2^{-1} I_2' = \begin{pmatrix} 2 & -\frac{1}{2} \\ 1 & -\frac{1}{3} \end{pmatrix} I_2'$$

$$A_3 = \begin{pmatrix} -1 & 4 \\ -2 & 0 \end{pmatrix} = U_3 A_3' Q_3 = \begin{pmatrix} -1 & 0 \\ -2 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 8 \end{pmatrix} \begin{pmatrix} 1 & -4 \\ 0 & 1 \end{pmatrix} = U_3 L_3$$

$$I = L_3^{-1} I_3' = \begin{pmatrix} 1 & \frac{1}{2} \\ 0 & \frac{1}{8} \end{pmatrix} I_3'$$

The gcd $\mathcal{L}$ of the lattices $\mathcal{L}_1, \mathcal{L}_2$ and $\mathcal{L}_3$ spanned respectively by the column-vectors of L_1^{-1}, L_2^{-1} and L_3^{-1} is spanned by the column-vectors of L^{-1}:

$$L^{-1} = \begin{pmatrix} \frac{1}{10} & 0 \\ \frac{1}{40} & \frac{1}{24} \end{pmatrix}$$

The expansion for which the non-periodicity condition is satisfied for all the vertices of P_3 is then given by

$$\begin{pmatrix} i \\ j \end{pmatrix} = \begin{pmatrix} \frac{1}{10} & 0 \\ \frac{1}{40} & \frac{1}{24} \end{pmatrix} \begin{pmatrix} i' \\ j' \end{pmatrix},$$

and the resulting polyhedron P_3' with an expanded variable space is:

$$P_3' = \begin{cases} i' - j' + 16n - 32m - 8 \geq 0 \\ j' + 36m - 18 \geq 0 \\ -i' + 60n + 40m + 125 \geq 0 \end{cases}$$

□

The number of integer points in the polyhedron P' resulting from the variable expansion is given by a non-periodic Ehrhart polynomial $\mathcal{E}_{P'}(N)$, i.e., a polynomial. But, as the variable space is expanded by L^{-1} of determinant $1/g$, the number of integer points in P' is roughly g times too big. In next subsection, we evaluate the error done by considering that the number of integer points is multiplied by g. Finally, an approximation of the number of integer points in P is given by $\frac{\mathcal{E}_{P'}(N)}{g}$.

Example 4.5. The number of integer points in P_3 is given by its Ehrhart polynomial:

$$\mathcal{E}_{P_3}(N) = \frac{361}{30}n^2 + \frac{209}{15}nm + \frac{121}{30}m^2 + \frac{1007}{30}n + \frac{583}{30}m +$$

$$
\begin{bmatrix}
23 & \frac{353}{15} & 23 & \frac{117}{5} & \frac{356}{15} & 23 & \frac{116}{5} & \frac{70}{3} & \frac{117}{5} & \frac{117}{5} & \frac{70}{3} & \frac{116}{5} & 23 & \frac{356}{15} & \frac{117}{5} \\
\frac{117}{5} & 23 & \frac{353}{15} & 23 & \frac{117}{5} & \frac{356}{15} & 23 & \frac{116}{5} & \frac{70}{3} & \frac{117}{5} & \frac{117}{5} & \frac{70}{3} & \frac{116}{5} & 23 & \frac{356}{15} \\
\frac{356}{15} & \frac{117}{5} & 23 & \frac{353}{15} & 23 & \frac{117}{5} & \frac{356}{15} & 23 & \frac{116}{5} & \frac{70}{3} & \frac{117}{5} & \frac{117}{5} & \frac{70}{3} & \frac{116}{5} & 23 \\
23 & \frac{356}{15} & \frac{117}{5} & 23 & \frac{353}{15} & 23 & \frac{117}{5} & \frac{356}{15} & 23 & \frac{116}{5} & \frac{70}{3} & \frac{117}{5} & \frac{117}{5} & \frac{70}{3} & \frac{116}{5} \\
\frac{116}{5} & 23 & \frac{356}{15} & \frac{117}{5} & 23 & \frac{353}{15} & 23 & \frac{117}{5} & \frac{356}{15} & 23 & \frac{116}{5} & \frac{70}{3} & \frac{117}{5} & \frac{117}{5} & \frac{70}{3} \\
\frac{70}{3} & \frac{116}{5} & 23 & \frac{356}{15} & \frac{117}{5} & 23 & \frac{353}{15} & 23 & \frac{117}{5} & \frac{356}{15} & 23 & \frac{116}{5} & \frac{70}{3} & \frac{117}{5} & \frac{117}{5} \\
\frac{117}{5} & \frac{70}{3} & \frac{116}{5} & 23 & \frac{356}{15} & \frac{117}{5} & 23 & \frac{353}{15} & 23 & \frac{117}{5} & \frac{356}{15} & 23 & \frac{116}{5} & \frac{70}{3} & \frac{117}{5} \\
\frac{117}{5} & \frac{117}{5} & \frac{70}{3} & \frac{116}{5} & 23 & \frac{356}{15} & \frac{117}{5} & 23 & \frac{353}{15} & 23 & \frac{117}{5} & \frac{356}{15} & 23 & \frac{116}{5} & \frac{70}{3} \\
\frac{70}{3} & \frac{117}{5} & \frac{117}{5} & \frac{70}{3} & \frac{116}{5} & 23 & \frac{356}{15} & \frac{117}{5} & 23 & \frac{353}{15} & 23 & \frac{117}{5} & \frac{356}{15} & 23 & \frac{116}{5} \\
\frac{116}{5} & \frac{70}{3} & \frac{117}{5} & \frac{117}{5} & \frac{70}{3} & \frac{116}{5} & 23 & \frac{356}{15} & \frac{117}{5} & 23 & \frac{353}{15} & 23 & \frac{117}{5} & \frac{356}{15} & 23 \\
23 & \frac{116}{5} & \frac{70}{3} & \frac{117}{5} & \frac{117}{5} & \frac{70}{3} & \frac{116}{5} & 23 & \frac{356}{15} & \frac{117}{5} & 23 & \frac{353}{15} & 23 & \frac{117}{5} & \frac{356}{15} \\
\frac{356}{15} & 23 & \frac{116}{5} & \frac{70}{3} & \frac{117}{5} & \frac{117}{5} & \frac{70}{3} & \frac{116}{5} & 23 & \frac{356}{15} & \frac{117}{5} & 23 & \frac{353}{15} & 23 & \frac{117}{5} \\
\frac{117}{5} & \frac{356}{15} & 23 & \frac{116}{5} & \frac{70}{3} & \frac{117}{5} & \frac{117}{5} & \frac{70}{3} & \frac{116}{5} & 23 & \frac{356}{15} & \frac{117}{5} & 23 & \frac{353}{15} & 23 \\
23 & \frac{117}{5} & \frac{356}{15} & 23 & \frac{116}{5} & \frac{70}{3} & \frac{117}{5} & \frac{117}{5} & \frac{70}{3} & \frac{116}{5} & 23 & \frac{356}{15} & \frac{117}{5} & 23 & \frac{353}{15} \\
\frac{353}{15} & 23 & \frac{117}{5} & \frac{356}{15} & 23 & \frac{116}{5} & \frac{70}{3} & \frac{117}{5} & \frac{117}{5} & \frac{70}{3} & \frac{116}{5} & 23 & \frac{356}{15} & \frac{117}{5} & 23
\end{bmatrix}_{n,m}
$$

Its approximation, given by the Ehrhart polynomial of the expanded polyhedron P_3' divided by $det(L) = 240$ is:

$$\frac{\mathcal{E}_{P_3'}(N)}{240} = \frac{361}{30}n^2 + \frac{209}{15}nm + \frac{121}{30}m^2 + \frac{1273}{40}n + \frac{737}{40}m + \frac{505}{24}$$

$\square$

4.6.3 Approximation error

In example 4.5, notice that all the terms of degree 2 (i.e., the terms for which the sum of the degrees along each variables equals 2) of the approximation $\mathcal{E}_{P_3'}(N)$ equal the terms of degree 2 of the original Ehrhart polynomial. In this section, we show that this is always the case:

Theorem 4.4. *Let $\mathcal{E}(N)$ be the Ehrhart polynomial of a n-dimensional polyhedron P. The terms of degree n of the approximation $\mathcal{E}'(N)$ by variable expansion equal the terms of degree n of $\mathcal{E}(N)$.*

Proof. The space $\mathcal{S}$ of the variables can be partitioned/paved into *unit cells* defined by

$$\mathcal{C}(I_0) = \{I \mid I_0 \le I < I_0 + \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}\}, I_0 \in \mathbb{Z}^n$$

where the $\le$ and $<$ operators are element-wise. Each unit cell contains one and only one integer point. When $\mathcal{S}$ is expanded by the transformation $I = L^{-1}I'$, giving the expanded space $\mathcal{S}'$, each unit cell is transformed into an *expanded cell* given by:

$$\mathcal{C}'(I_0) = \{I' \mid I_0 \le L^{-1}I' < I_0 + \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}\},$$

which contains exactly $g = det(L)$ integer points.

If the polyhedron P could be partitioned exactly into whole unit cells (i.e., if it is a hyper-rectangle with integer vertices), then its image by expansion P' would be the union of the image of the unit cells of P. Then, its number of integer points would be exactly g times the number of integer points in P. But in the general case, some unit cells are partly inside P and partly outside. Two cases can be distinguished:

- a unit cell with $I_0 \notin P$ but having a non-empty intersection with P: the corresponding *expanded cell* may contain integer points that belong to P. As these integer points do not correspond to an integer point of P, they are *extra points* which tend to lead to an over-approximation of $\mathcal{E}(N)$.

- a unit cell with $I_0 \in P$ but not being completely included in P: the corresponding *expanded cell* may contain integer points that do not belong to P. As these integer points do correspond to an integer point of P, they are *missing points* which tend to lead to an under-approximation of $\mathcal{E}(N)$.

Example 4.6. The unit cell $\mathcal{C}(0,0) = \{0 \le i < 1; 0 \le j < 1\}$ and the polyhedron $\{3i + 4j - 2 \ge 0\}$ in $\mathbb{Z}^2$ are represented in figure 4.1. They have a non-empty intersection. Expanding the variable space by $\begin{pmatrix} i \\ j \end{pmatrix} = L^{-1} \begin{pmatrix} i' \\ j' \end{pmatrix}$ with $L = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ gives the expanded unit cell $\mathcal{C}'(0,0) = \{0 \le i' + j' < 2; 0 \le i' - j' < 2\}$ and the expanded polyhedron $P' = \{7i' - j' - 4 \ge 0\}$ represented in figure 4.2.

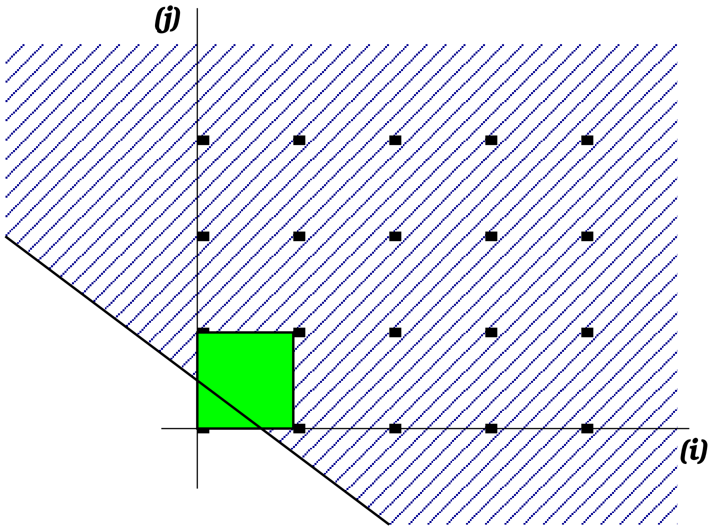

Figure 4.1: A unit cell having a non-empty intersection with a polyhedron

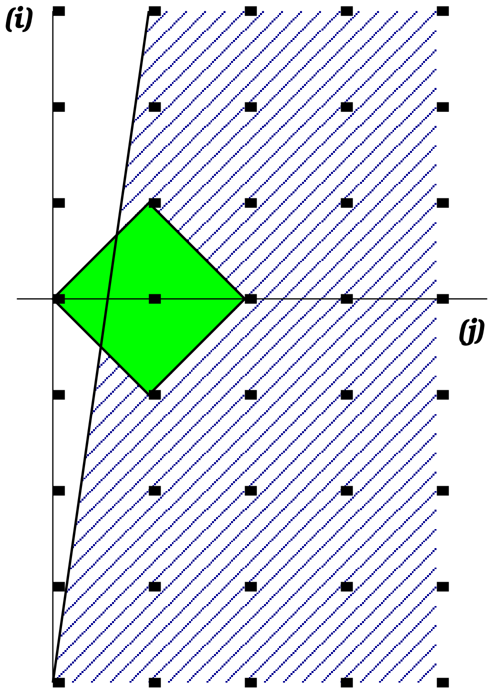

Figure 4.2: The expanded unit cell and polyhedron, with an extra integer point

□

We then have:

$$\mathcal{E}(N) = \mathcal{E}'(N) + (nmp(N) - nep(N))/g,$$

where nmp and nep stand respectively for *number of missing points* and *number of extra points*. The approximation error ξ is then given by:

$$\xi = |(nmp(N) - nep(N))/g| \tag{4.8}$$

This difference is of course periodic, but now we want to show that its degree is strictly inferior to the maximum degree of $\mathcal{E}(N)$. Notice that the extra and missing points are somehow similar: we give an upper bound on the number of extra points, knowing that the upper bound on the missing points is similar.

Extra integer points in P' are the image of some rational points of a unit cell with $I_0 \notin P$ but having a non-empty intersection with P. So the integer points I_0 that are somehow close to P can lead to extra points in P': at most $g-1$ points. The set $\mathbb{I}_0$ of such integer points I_0 belong to a finite set of hyperplanes parallel to the faces of P. Let P defined by the set of m constraints:

$$f(N) = \{ \bigcap_{j \in [1..m]} f_j(I,N) \geq 0\}, \tag{4.9}$$

where $f_j(I,N) = \sum_k a_k^j i_k + \sum_l b_l^j n_l + c^j \geq 0, j \in [1..m]\}$. The condition for a unit cell to intersect P is that at least one of its vertices satisfies (4.9). The general form of a vertex of $\mathcal{C}(I_0)$ is $I_0 + Q$, where $Q = (q_k) \in \mathbb{Z}^n$ and q_k can be 0 or 1. Notice that the case $Q = (0)$ is excluded as the considered I_0 must not belong to P. So the general condition on I_0 for $\mathcal{C}(I_0)$ to have a potential intersection with P is:

$$I_0 \in \{\{I \mid I + Q \in P\} \setminus \{I \in P\}\}$$

$$\Leftrightarrow I_0 \in \{I \mid \sum_k a_k^j (i_k + q_k) + \sum_l b_l^j n_l + c^j \geq 0\} \setminus \{I \mid \sum_k a_k^j i_k + \sum_l b_l^j n_l + c \geq 0\}$$

$$\Leftrightarrow I_0 \in \{I \mid \sum_k a_k^j i_k + \sum_l b_l^j n_l + (\sum_k a_k^j q_k + c^j) \geq 0\} \setminus \{I \mid \sum_k a_k^j i_k + \sum_l b_l^j n_l + c \geq 0\}.$$

The set of all the possible values of Q gives a disjunctive set of inequalities. Considering a given inequality $\sum_k a_k^j i_{0,k} + \sum_l b_l^j n_l + (\sum_k a_k^j q_k + c^j) \geq 0$ (i.e., a given value of j), all the inequalities (for all the possible values of Q) are redundant with (i.e., they imply) the inequality for which the value of $(\sum_k a_k^j q_k + c)$ is maximal. This value is maximum if $q_k = 1$ when a_k is nonnegative: the maximum value is then $(\sum_k a_k^{j+} + c)$, where

$$a_k^{j+} = \begin{cases} a_k^j \ if \ a_k^j \geq 0 \\ 0 \ if \ a_k^j < 0 \end{cases}$$

A looser but simpler necessary condition for $\mathcal{C}(I_0)$ to intersect P can then be derived: $I_0 \in \Upsilon$, where

$$\Upsilon = \{I \mid \sum_k a_k^j i_k + \sum_l b_l^j n_l + \sum_k a_k^{j+} + c^j \geq 0\} \setminus \{I \mid \sum_k a_k^j i_k + \sum_l b_l^j n_l + c \geq 0\},$$

which can also be written :

$$\Upsilon = (\bigcap_{j \in [1..m]} \{f_j(I,N) + \alpha_j \geq 0\}) \setminus (\bigcap_{j \in [1..m]} f_j(I,N) \geq 0\}),$$

where $\alpha_j = \sum_k a_k^{j+}$. Using the set theory (in particular, the formula $A \setminus B = A \cap \bar{B}$), Υ can be reduced to a union of m convex polyhedra:

$$\Upsilon = \bigcup_{r \in [1..m]} (\bigcap_{j \in [1..m]} \{f_j(I,N) + \alpha_j \geq 0\} \cap \{f_r(I,N) < 0\}) = \bigcup_{r \in [1..m]} \Upsilon_r$$

For a given convex polyhedron Υ_r of the union, there is always a couple of inequalities $f_r(I,N) + \alpha_r \geq 0$ and $f_r(I,N) < 0$.

Then, we can say that $\mathcal{C}(I_0)$ may lead to extra integer points in P' if I_0 is an integer solution to the equation

$$f_r(I,N) + \beta = 0, \beta \in \mathbb{Z}$$

with $0 < \beta \leq \alpha_r$.

The number of $(n-1)$-dimensional polytopes (the solution belongs to Υ_r) defined by this equality is less than or equal to the number of distinct values for β, i.e.. α_r. The number of integer solutions for I_0 in a $(n-1)$-dimensional polytope for a given value of β is given by an Ehrhart polynomial $\mathcal{E}_{\Upsilon_r,\beta}(N)$, whose degree is $(n-1)$.

Finally, the number of extra integer points in P' is less than or equal to the number of unit cells that can lead to extra points times the number of integer points in the expanded unit cell. Formally, it is less than or equal to :

$$nep(N) \leq \sum_{r=1}^{m} (\sum_{\beta=1}^{\alpha_r} (\mathcal{E}_{\Upsilon_r,\beta}(N))) \times (g-1)$$

As neither of m, α_r and g depend on N, the degree of $nep(N)$ is the maximum degree of the different $\mathcal{E}_{\Upsilon_{r,\beta}}(N)$, i.e., $n-1$.

Obviously, a very similar reasoning can be applied to the number of missing points: $nmp(N)$ is less than or equal to a sum of Ehrhart polynomials whose degrees are $n-1$. According to equation (4.8), the approximation error for $\mathcal{E}(P)$ using the approximation by variable expansion is an Ehrhart polynomial whose maximum degree can be $n-1$.

This proof can be extended to a non-full-dimensional polyhedron by using the transformation presented in section 2.5.3, which transforms a polyhedron $P \in \mathbb{Q}^{n+n'}$ whose n-dimensional affine hull intersects an integer lattice $\mathcal{L}_1$ into a full-dimensional $\mathbb{Z}$-polyhedron $P' \cap \mathbb{Z}^n$. $\square$

4.7 Computing the Ehrhart polynomial of an expanded polyhedron

To simplify the following discussion, we will consider a full-dimensional rational parametric polyhedron P. Extension to a non-full-dimensional polyhedron is done as usual (across this thesis). We consider a polyhedron P resulting from a variable space expansion, as described in previous section.

We want to compute the Ehrhart polynomial of P, by using generating functions, similarly as in the Seghir-Verdoolaege algorithm [77, 83]. However, we want to take advantage of the special properties of P.

Several researchers have worked on generating functions as a way to enumerate the integer points in a polytope. An article by Barvinok and Pommersheim [6] makes a clear inventory of the known results in this theory. The generating function of $P \cap \mathbb{Z}^n$, the integer points of P is a Laurent series defined by:

$$f(P; x) = \sum_{I \in P \cap \mathbb{Z}^n} x^I,$$

where x and I are n-dimensional, and the vectorial notation $x^m = x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ is used, as in section 4.2. The Ehrhart polynomial of P is then given by $f(P; (1 \cdots 1))$. The power of Barvinok's algorithm, on which the Seghir-Verdoolaege algorithm is based, relies on properties of the generating functions:

(a) the generating function of P equals the sum of the generating functions of its cones (Brion's theorem),

(b) for any convex polyhedral cone K, the series $f(K;x)$ is a quotient of two non-zero Laurent polynomials.

In particular, if the n rays u_k of K, pointed on a vertex v, form a unimodular basis, we have $f(K;x) = x^v \Pi_{k=1}^n \frac{1}{(1-x^{u_k})}$. Then, known techniques allow to evaluate $f(K;x)$ for $x = (1 \cdots 1)$, giving the Ehrhart polynomial of P, $\mathcal{E}_P(N) = \sum_i f(K_i;x)$.

Another important property of the generating functions is the following:

Property 1. *$f(P;x) = 0$ if P contains a straight line.*

By the way, this property lead to an alternative proof [6, 47] of Brion's theorem. The algorithm we propose is mainly based on this last property. As Barvinok, we can compute the generating function for each cone of P and then sum the generating functions of the different cones.

Consider a rational parametric polyhedron P resulting from a variable space expansion. By construction, it is the intersection of *unimodular constraint-cones*, and its vertices are integer.

Definition 5 (m-constraint-cone). *A m-*constraint-cone* K in a n-dimensional space is defined by the intersection of m constraints. It is a cone pointed on a vertex v, written a set of inequalities:*

$$AI + BN + C \geq 0,$$

where A, B and C are $m \times n$, $m \times p$ and $m \times 1$ matrices. A unimodular constraint-cone *is a (simplicial) n-constraint-cone such that A is unimodular. The vertex v of K saturates all the inequalities (i.e., v is solution to the corresponding equalities).*

□

Each constraint-cone K of P of vertex v is the intersection of unimodular constraint-cones:

$$K = \bigcap_i \mathcal{U}_i.$$

A constraint-cone is equivalent to a cone written as the positive combination of rays r_j and v (Minkowski representation):

$$\mathcal{C} = v + \sum_{j=1}^{n} \alpha_j r_j, \alpha_j \geq 0, r_j \in \mathbb{Z}^n.$$

r_j is called a ray of $\mathcal{C}$.

Let two constraint-cones $\mathcal{C}_1$ and $\mathcal{C}_2$ of vertex v such that $u \in \mathbb{Z}^n$ is a ray of $\mathcal{C}_1$ and $-u$ is a ray of $\mathcal{C}_2$:

$$\mathcal{C}_1 = v + \alpha_1 u + \sum_{j=2}^{n} \alpha_j r_{1,j}, \alpha_j \geq 0,$$

$$\mathcal{C}_2 = v - \alpha_2 u + \sum_{j=2}^{n} \alpha_j r_{2,j}, \alpha_j \geq 0,$$

The generating function of $\mathcal{C}_1 \cap \mathcal{C}_2$ is given by:

$$f(\mathcal{C}_1 \cap \mathcal{C}_2; x) = f(\mathcal{C}_1; x) + f(\mathcal{C}_2; x) - f(\mathcal{C}_1 \cup \mathcal{C}_2; x).$$

$\mathcal{C}_1 \cup \mathcal{C}_2$ can be written as a positive combination of rays and v:

$$\mathcal{C}_1 \cup \mathcal{C}_2 = v + (\alpha_1 - \alpha_2)u + \sum_{j=1}^{n} \alpha_j r_{1,j} + \sum_{j=1}^{n} \alpha_k r_{2,j}, \alpha_j \geq 0 \alpha_k \geq 0. \tag{4.10}$$

As $\{(\alpha_1 - \alpha_2)u, \alpha_1 \geq 0, \alpha_2 \geq 0\}$ defines a straight line, we have by property 1:

$$f(\mathcal{C}_1 \cup \mathcal{C}_2; x) = 0,$$

which gives the following property:

Property 2 (composability). *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two constraint-cones of vertex v. If u is a ray of the constraint-cone $\mathcal{C}_1$ and $-u$ is a ray of the constraint-cone $\mathcal{C}_2$, we have:*

$$f(\mathcal{C}_1 \cap \mathcal{C}_2; x) = f(\mathcal{C}_1; x) + f(\mathcal{C}_2; x).$$

$\mathcal{C}_1$ and $\mathcal{C}_2$ are said to be composable.

A m-constraint-cone can be defined as the intersection of C_m^n unimodular constraint-cones [1]. But these unimodular constraint-cones are not in general composable with each

[1]Here, C_m^n denotes the number of combinations of n inequalities among m

other. We must find a definition of K as the intersection of unimodular constraint-cones $\mathcal{C}_i$ that are composable with each other. The generating function of K is then the sum of the generating functions of $\mathcal{C}_i$.

A simple divide-and conquer algorithm leads to this decomposition. As constraints incident to a cone are split away from each other, similarly as the banana skin is split from one end of the banana, we call it the *banana skin* algorithm.

```
C: m-constraint cone

decomp(C) {
if C is a n-constraint-cone, return C.
B: n − 1 inequalities of C
Compute the m − n unimodular constraint-cones C' of C made of:
- the n − 1 chosen inequalities
- one of the other m − n inequalities
u: basis vector of the kernel of B
S_1: set of unimodular constraint-cones of C' having the ray u
S_2: set of unimodular constraint-cones of C' having the ray −u
return decomp(S_1), decomp(S_2)
}
```

Each call of *decomp($\mathcal{C}$)* splits the m-constraint-cone $\mathcal{C}$ into two composable constraint-cones. The rays for a unimodular constraint-cone are usually obtained by inverting the matrix A, which is n-dimensional. For a m-constraint-cone, $m-n$ inversions would be needed. However, u needs to be determined once only, for instance using a Hermite normal form of B. Then, we just need the sign of the ray αu, as its direction is (also) determined by B. Let u' be the vector chosen among the other $(m-n)$ inequalities. The sign of α is determined by $\alpha u.u' = 1$.

Obviously, at most $n-m$ splittings occur for a m-constraint-cone. At each splitting, the bulk of the computation is in the Hermite normal form. The complexity of such a recursive splitting is clearly lower than Barvinok's decomposition into simplicial unimodular cones.

The approximation method by variable space expansion, associated with the constraint-cone splitting algorithm presented here, is then expected to be the fastest approximation algorithm.

4.8 Other approximation algorithms

We have devised polynomial-time (for fixed dimension) algorithms for computing an approximation having the following properties:

a. its approximation error is of maximal degree $d-1$, where d is the geometric dimension of P. Then, the asymptotic limit of this error is zero when the Ehrhart polynomial is of degree d.

b. it is valid on the whole validity domain of the Ehrhart polynomial they correspond to.

Now let us compare different approaches to ours.

4.8.1 Interpolation by a polynomial

We have seen that the Ehrhart polynomial of P is a polyhedron if we restrict the values of the parameters N to a certain lattice (so that the non-periodicity condition is fulfilled for P). Computing a polynomial by interpolation on such a lattice is the idea of the parameter space compression method coupled with the Clauss-Loechner-Wilde algorithm. Unless the interpolation is done on points on such a lattice, there is no guarantee on the approximation error on the whole validity domain (but only on the domain covered by the interpolation points). Considering this, it is important to note that validity domains can be unbounded.

Example 4.7. To illustrate a simple unlucky interpolation, consider the following polytope: $\{1 \leq 4i \leq n\}$, where n is the parameter. The Ehrhart polynomial of P is $\mathcal{E}_P(N) = \frac{1}{4}n + \begin{bmatrix} 0 & -\frac{1}{4} & -\frac{1}{2} & -\frac{3}{4} \end{bmatrix}_n$. P is one-dimensional, so it can be inteprolated by a polynomial of degree one. Assume the chosen interpolation points are $(n = 1; n = 7)$. The interpolated polynomial is then $\mathcal{E}'(n) = \frac{1}{6}n - \frac{1}{6}$. The absolute approximation error is asymptotically $\frac{1}{12}n$, the relative error being $\frac{1}{3}$.

□

4.8.2 Average value of the coefficients

Taking the average value of the coefficients, as proposed by Heine and Slovik [42], gives an approximation whose error is asymptotically zero only in the case where the coefficient of highest degree of the Ehrhart polynomial is non-periodic. Notice that this method needs to compute the Ehrhart polynomial first. The periodic character of Ehrhart polynomials seems to be the main problem: a non-periodic polyhedron is obviously easier to manipulate.

4.9 Conclusion

In this chapter, we have extended Ehrhart's conjecture, giving a more precise relation between the coefficients of the constraints defining P and the periodicity of its Ehrhart polynomial. These informations allow to transform P into a somehow similar polyhedron with a non-periodic Ehrhart polynomial. The application of this is two methods for approximating the Ehrhart polynomial of P by a non-periodic polynomial. Besides, a new efficient method for computing Ehrhart polynomials of polyhedra with particular properties has been shown. In future works, we will try to extend it to any kind of polyhedron.

Chapter 5

Implementation

A part of the concepts and algorithms developed in this thesis has been implemented. Still, some parts have been developed too recently to be implemented. Some other parts needed the clarity brought by this thesis to be implemented in a robust way. These parts are underway to be completed.

5.1 Periodics

5.1.1 Explicit periodics

In [66], we have studied periodic numbers, periodic polyhedra and Ehrhart polynomials as being instances of a generic class: the *periodics*. Periodics are defined as multidimensional functions of integer values over a monoid K. These functions have a fixed period along each of their n variables, so they can be represented as n-dimensional arrays. A generic class of periodics has been implemented using such a data structure, in $C++$. Generic operators, as well as iterators that are aware of the spatial locality of the contained data, have been written.

Instantiating K to integers, polyhedra and polynomials has firstly allowed us to manipulate easily periodic numbers (in their explicit form), periodic polyhedra and Ehrhart polynomials. This implementation of periodic numbers has been used by R. Seghir [77] in preliminary versions of his program for computing Ehrhart polynomials by a parametric extension of Barvinok's algorithm. A cast operator, from this class of explicit periodic numbers to the periodic numbers under the Polylib representation, has been written for this purpose. This usage has been obsoleted later, when Seghir and Verdoolaege [77, 83] put their software contributions together and focused on using the symbolic form for periodic numbers in order to avoid the exponential complexity of the explicit form.

This generic class has also been instantiated with polynomials in an attempt of Ehrhart polynomials *pretty printing* in LaTeX. This attempt was motivated by the periodic need of writing down Ehrhart polynomials in scientific articles, and by the not-so-comfortable existing text output. A cast operator from the Polylib representation to the form of a periodic polynomial has been written for this purpose, as well as a pretty printer for periodic polynomials.

A redundancy problem showed up when using this class for representing periodic polyhedra, as the non-periodic part can be significant. A more efficient way to represent a periodic polyhedron with m constraints is then to use a matrix with m rows and a vector of m periodic numbers.

The advantage of this generic class is that it is possible to use new instances of periodics quickly. However, as the explicit periodic form is inherently exponential, the periodic class developed this way is likely to take significant amounts of memory space. Moreover, the readability of the periodic may become easily bad.

The need for a symbolic form of periodic numbers has showed up very quickly for these reasons, but also for the known performance reasons. This form is presented in next section.

5.1.2 Symbolic periodics

The symbolic form of the periodic numbers is defined by the way they can be computed. Roughly speaking, periodic numbers we use are nested modulos of affine functions of the variables. When constructing pseudo-facets, the modulo of a periodic affine function is computed, which defines the periodic supporting hyperplane. When computing the corresponding projected pseudo-facet, linear combinations between the periodic affine constraints are done. Besides, the most simple form of a periodic number is a rational number. As we only deal with integers in our implementations, it is (for us) an integer number. The symbolic form of a periodic number can then be summed up by the following context-free grammar:

< pn > ⟶ < integer>
| <integer> * <pn> + <integer> * <pn>
| (<f(I)> + <pn>) mod <integer>

where <f(I)> is an affine function of the variables, <integer> is an integer constant. The three derivation alternatives define the three C++ constructors we use for a symbolic periodic number. We have also written a cast operator for converting a periodic number in symbolic form into its explicit form.

This symbolic periodic number class is used, as expected in our pseudo-vertices computation function, presented in next section.

5.1.3 Pseudo-vertices

The periodic polyhedron representation is made of a $m \times (n + p + 1)$ matrix, along with a m-dimensional vector of symbolic periodic numbers. The pseudo-vertices computation function builds either the whole pseudo-facets tree is we want to compute the integer hull. As explained in section 3.2, branches are selected (the others are pruned) if we look for the maximum w.r.t. a given order. The conversion operator into explicit periodic number can then be used to compute the maximal extremal points and the corresponding validity domain. Iterators over explicit periodic numbers can be linked together, so the periodic validity domain is easily linked to the periodic coordinates of the extremal point it corresponds to.

5.1.4 Periodic polyhedra as input

For now, our programs can only take a polyhedron as its input. In order to deal with problems whose input involve periodic polyhedra, we must develop a parser for periodic polyhedra. The context-free grammar presented in subsection 5.1.2 will be helpful to parse periodic numbers in their symbolic form.

5.2 Ehrhart polynomials approximation

The approximation by variable space expansion, presented in section 4.6, has been implemented for full-dimensional polyhedra, by modifying the Polylib function *Polyhedron_Enu-*

merate(). The function firstly computes the coordinates of each vertex of the considered polyhedron P, along with its validity domain. At this stage, we also compute the non-periodicity condition for the vertex, as an expansion matrix. Then, vertices are grouped into sets sharing a common validity domain. All these vertices will form a polyhedron, for which we compute the expansion matrix L, which is the Hermite normal form of the expansion matrices of the grouped vertices.

This expansion is then applied to the polyhedron, giving a polyhedron P'. The Ehrhart polynomial of P' is then computed (instead of the Ehrhart polynomial of P), and divided by $det(L)$.

Chapter 6

Conclusion

We have centered this thesis on the periodicity appearing when manipulating integer points in rational polyhedra, and shown the particular role of periodic polyhedra in this frame. First, the integer hull of a parametric polyhedron, for which we give a computation algorithm, is a periodic polyhedron. Furthermore, in the field of parametric integer linear programming, which is frequently used in the polytope model, the integer maximum w.r.t. a (hierarchically) linear order is a set of periodic polyhedra. This result is corroborated by a well-known particular case, the lexicographic extremum of a parametric polyhedron P: this extremum is a pseudo-vertex, i.e., a 0-dimensional periodic polyhedron. We believe that some optimizations still have to be found for our parametric integer linear programming algorithm. PIP, the library dedicated to computing this particular extremum, will be used as an indicator in our future performance measurements.

Besides, we would like to extend the concept of pseudo-facet w.r.t. a variable to pseudo-facets w.r.t. a vector. This might enable new profitable degree of freedom for computing the solution, and could also reduce the complexity of the algorithm.

The periodicity of the integer hull explains the periodic character of Ehrhart polynomials. We use this link between the integer hull of P and its Ehrhart polynomial to extend Ehrhart's conjecture, giving a less restrictive condition on the polyhedron P to have a non-periodic Ehrhart polynomial. We derive some side results from this new non-periodicity condition: two distinct approximation methods for Ehrhart polynomials, an optimization for the Clauss-Loechner-Wilde interpolation method for computing Ehrhart polynomials, and a new method for computing Ehrhart polynomials in particular cases.

We wish to extend this technique, based on generating functions as in the Barvinok-like algorithms, to the general case in future work.

The algorithms presented in chapter 2 and 3 have been presented with non-periodic polyhedra. But they also work for periodic polyhedra, as they manipulate pseudo-facets, which are periodic polyhedra. Actually, the polytope model suffers from a major limitation: some of the algorithms working on polyhedra produce a result that is a periodic polyhedron. Then, further operations on polyhedra may not be directly applicable to this periodic polyhedron. This problem raises the need for a library that manipulates periodic polyhedra, in the manner of Polylib, for instance.

Code generation is another classical problem in the polytope model. We also wish to solve it using periodic polyhedra and the pseudo-facets machinery, and to evaluate this solution.

Bibliography

[1] K. Aardal. Lattice basis reduction and integer programming. Technical Report UU-CS 1999-37, Utrecht University: Information and Computing Sciences, 1999.

[2] K. Aardal and S. van Hoesel. Polyhedral techniques in combinatorial optimization i: Theory. *Statistica Neerlandica*, 50:3–26, 1996.

[3] K. Aardal and S. van Hoesel. Polyhedral techniques in combinatorial optimization ii: Computations. *Statistica Neerlandica*, 50:3–26, 1996.

[4] C. Ancourt and F. Irigoin. Scanning polyhedra with DO loops. In *Proc. ACM SIGPLAN '91*, pages 39–50, june 1991.

[5] A. Barvinok. A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Mathematics of Operations Research*, 19:769–779, 1994.

[6] A. Barvinok and J. Pommersheim. An algorithmic theory of lattice points in polyhedra. *New perspectives in Algebraic combinatorics*, 38:91–147, 1999.

[7] A. Bockmayr and F. Eisenbrand. Cutting planes and the elementary closure in fixed dimension. *Mathematics of Operations Research*, 26(2):304–312, 2001.

[8] P. Boulet and P. Feautrier. Scanning polyhedra without Do-loops. In *Int. Conf of Parallel Architectures and Compilation Techniques PACT'98*. IEEE, October 1998.

[9] P. Boulet and X. Redon. Communication pre-evaluation in hpf. Research Report 93-36, LIP, Ecole Normale Supérieure de Lyon, France, November 1998.

[10] J. Bourgeois, F. Spies, and M. Tréhel. Performance prediction of distributed applications running on a network of workstations. In *Parallel and Distributed Applications and Activities: PDPTA'99*, pages 672–678, Las Vegas, USA, June 1999.

[11] V. Braberman, D. Garbervetsky, and S. Yovine. On synthesizing parametric specifications of dynamic memory utilization. technical report TR-2004-03, VERIMAG, 2004.

[12] N.V. Chernikova. Algorithm for discovering the set of all the solutions of a linear programming problem. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 6(8):282–293, 1968.

[13] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4:305–337, 1973.

[14] Ph. Clauss. *Synthèse d'algorithmes systoliques et implantation optimale en place sur réseaux de processeurs synchrones.* PhD thesis, Université de Franche-Comté, May 1990.

[15] Ph. Clauss. The volume of a lattice polyhedron to enumerate processors and parallelism. Technical Report 95-11, ICPS, 1995.

[16] 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.

[17] Ph. Clauss. Handling memory cache policy with integer points countings. In Lengauer, Griebl, and Gorlatch, editors, *Euro-Par'97, Passau*, pages 285–293. Springer-Verlag, LNCS 1300, August 1997.

[18] Ph. Clauss. Méthodes polyédriques pour la parallélisation et l'optimisation de programmes. Habilitation à diriger des recherches, Université Louis Pasteur, Habilitation à diriger des recherches, 2000. http://icps.u-strasbg.fr/.

[19] Ph. Clauss and V. Loechner. Parametric analysis of polyhedral iteration spaces. In Fortes, Mongenet, Parhi, and Taylor, editors, *IEEE Int. Conf. on Application Specific Array Processors, ASAP'96*, pages 415–424. IEEE Computer Society Press, August 1996.

[20] Ph. Clauss, V. Loechner, and B. Meister. Minimizing strides in loops with affine array references. In *CPC 2001, 9th Workshop on Compilers for Parallel Computers*, June 2001.

[21] Ph. Clauss, V. Loechner, and D. K. Wilde. Deriving formulae to count solutions to parameterized linear systems using Ehrhart polynomials: Applications to the analysis of nested-loop programs. Technical Report 97-05, ICPS, 1997.

[22] Ph. Clauss and B. Meister. Automatic memory transformations to optimize spatial locality in parameterized loop nests. *ACM SIGARCH*, 28(1):11–19, March 2000.

[23] J.-F. Collard. Space-time transformation of while-loops using speculative execution. In IEEE, editor, *proc. of the Scalable High-Performance Computing Conf.(SHPCC'94)*, pages 429–436, May 1994.

[24] J.-F. Collard, P. Feautrier, and T. Risset. Construction of DO loops from systems of affine constraints. *Parallel Processing Letters*, 5(3):421–436, 1995.

[25] G.B. Dantzig. Maximization of a linear function of variables subject to linear inequalities. In T.C. Koopmans (ed.), editor, *Activity Analysis of Production and Allocation (Cowles Commission Monograph No. 13)*, chapter 21, pages 339–347. Wiley, 1951.

[26] E. Ehrhart. *Polynômes arithmétiques et Méthode des Polyèdres en Combinatoire*, volume 35 of *International Series of Numerical Mathematics*. Birkhäuser Verlag, Basel/Stuttgart, 1977.

[27] R. A. Van Engelen, K. Gallivan, and B. Walsh. Parametric timing estimation with the Newton-Gregory formulae. *Journal of Concurrency and Computation: Practice and Experience*, 2004. Accepted for publication in 2004.

[28] Peter Faber, Martin Griebl, and Christian Lengauer. A closer look at loop-carried code replacement. In *Proc. GI/ITG PARS'01*, PARS-Mitteilungen Nr.18, pages 109–118. Gesellschaft für Informatik e.V., November 2001.

[29] P. Feautrier. Parametric integer programming. *RAIRO Recherche Opérationnelle*, 22:243–268, September 1988.

[30] P. Feautrier. Dataflow analysis of scalar and array references. *Int. J. of Parallel Programming*, 20(1):23–53, February 1991.

[31] P. Feautrier. Some efficient solutions to the affine scheduling problem, part 1 : one dimensional time. *Int. J. of Parallel Programming*, 21(5):313–348, October 1992.

[32] P. Feautrier. Lncs 1132. In *The Data Parallel Programming Model*, chapter Automatic parallelization in the Polytope Model, pages 79–100. Springer, 1996.

[33] P. Feautrier, J. Collard, and C. Bastoul. Solving systems of affine (in)equalities. Technical report, PRiSM, Versailles University, 2002.

[34] F. Fernandez and P. Quinton. Extension of Chernikova's algorithm for solving general mixed linear programming problems. Technical Report 437, IRISA, Rennes, France, 1988.

[35] J. Ferrante, J. Sarkar, and W. Thrash. On estimating and enhancing cache effectiveness. In *Advances in Languages and Compilers for Parallel Processing*, pages 328–343. MIT Press, 1991.

[36] R. E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64:275–278, 1958.

[37] Martin Griebl, Paul A. Feautrier, and Christian Lengauer. Index set splitting. *Int. J. Parallel Programming*, 28(6):607–631, 2000.

[38] M. Grötschel, L. Lovász, and A. Schrijver. Geometric methods in combinatorial optimization. In W.R. Pulleybank, editor, *Progress In Combinatorial Optimization*, pages 167–183. Academic Press, 1984.

[39] F. Hannig and J. Teich. Design Space Exploration for Massively Parallel Processor Arrays. In Victor Malyshkin, editor, *Parallel Computing Technologies, 6th International Conference, PaCT 2001, Proceedings*, volume 2127 of *Lecture Notes in Computer Science (LNCS)*, pages 51–65, Novosibirsk, Russia, September 2001. Springer.

[40] F. Hannig and J. Teich. Energy Estimation of Nested Loop Programs. In *Proceedings 14th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA 2002)*, Winnipeg, Manitoba, Canada, August 2002. ACM Press.

[41] W. Harvey. *Integer Constraint Solving Methods*. PhD thesis, The University of Melbourne, January 1998.

[42] F. Heine and A. Slowik. Volume driven data distribution for numa-machines. In *EuroPar 2000 Parallel Processing*, pages 415–424. LNCS, 2000.

[43] H.W. Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8:538–548, 1983.

[44] R. D. Ju, J-F. Collard, and K. Oukbir. Probabilistic memory disambiguation and its application to data speculation. *SIGARCH Comput. Archit. News*, 27(1):27–30, 1999.

[45] M. Kandemir, A. Choudhary, J. Ramanujam, , and P. Banerjee. A matrix-based approach to global locality optimization. *Journal of Parallel and Distributed Computing*, 58:190–235, 1999. available online at http://www.idealibrary.com.

[46] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–396, 1984.

[47] M. Knudsen. Brion's theorem. Graduate project, University of Aarhus, April 2004. http://home.imf.au.dk/niels/aco.html.

[48] I. Kodukula, K. Pingali, R. Cox, and D. Maydan. An experimental evaluation of tiling and shackling for memory hierarchy management. In *Proceedings of the 13th international conference on Supercomputing*, pages 482–491. ACM Press, 1999.

[49] E. Kranich. Interior point methods for mathematical programming : A bibliography. Technical Report 171, P.O. Box 940, D–5800 Hagen 1, Germany, May 1991.

[50] B. Lasserre. The integer hull of a convex rational polytope. *Discrete and Computational Geometry*, 32(1):129–139, 2004.

[51] H. Le Verge. A note on chernikova's algorithm. Technical Report 635, IRISA, Rennes, France, 1992.

[52] C. Lengauer. Loop parallelization in the polytope model. In *CONCUR 93*, 1993.

[53] A.W. Lim and M.S. Lam. Communication-free parallelization via affine transformations. *K. Pingali et al., editors, Languages and Compilers for Parallel Computing,LNCS*, 892:92–106, 1995. Springer-Verlag.

[54] B. Lisper. Fully automatic, parametric worst-case execution time analysis. In Jan Gustafsson, editor, *Third International Workshop on Worst-Case Execution Time (WCET) Analysis*, pages 77–80, July 2003.

[55] 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/.

[56] V. Loechner. Polylib: A library for manipulating parameterized polyhedra. Technical report, ICPS, Strasbourg, France, 1999.

[57] V. Loechner, B. Meister, and Ph. Clauss. Data sequence locality: a generalization of temporal locality. In *EuroPar, European Conference on Parallel Computing*. MRCCS, August 2001.

[58] V. Loechner, B. Meister, and Ph. Clauss. Precise data locality optimization of nested loops. *The Journal of Supercomputing*, 21(1):37–76, January 2002. Kluwer Academic Pub.

[59] V. Loechner and C. Mongenet. Communication optimization for affine recurrence equations using broadcast and locality. *Int. J. of Parallel Programming*, 28(1), 2000.

[60] V. Loechner and D. K. Wilde. Parameterized polyhedra and their vertices. *International Journal of Parallel Programming*, 25(6):525–549, December 1997.

[61] J.A. De Lorea, R. Hemmecke, J. Tauzer, and R. Yoshida. Effective lattice point counting in rational convex polytopes. Technical report, University of California at Davis, March 2003.

[62] L. Lovász and H.E. Scarf. The generalized basis reduction algorithm. *Mathematics of Operations Research*, 17:751–764, 1992.

[63] P. McMullen. Lattice invariant valuations on rational polytopes. *Arch. Math.(Basel)*, 31:509–516, 1978/79.

[64] B. Meister. Localité de données dans les opérations stencil. In *RenPar'13*, pages 37–42. ASTI, april 2001. french text.

[65] B. Meister. Localité de données et compression mémoire dans les opérations stencil. Rapport de recherche, ICPS-LSIIT, Septembre 2001. En français.

[66] B. Meister. Using periodics in integer polyhedral problems. research report ICPS, LSIIT - ICPS, 2003. http://icps.u-strasbg.fr/∼ meister.

[67] B. Meister, V. Loechner, and Ph. Clauss. The polytope model for optimizing cache locality. Technical Report RR 00-03, ICPS-LSIIT, May 2000.

[68] M. Minoux. *Programmation Mathématique, théorie et algorithmes*, volume 2 of *Collection Technique et Scientifique des Télécommunications*. Dunod, 1983.

[69] 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.

[70] S. Nookala and T. Risset. A library for $\mathbb{Z}$-polyhedral operations. Technical report 1330, IRISA, May 2000. ISSN 1166-8687.

[71] M.F.P. O'Boyle and P.M.W. Knijnenburg. Nonsingular data transformations: Definition, validity, and applications. *Int. J. of Parallel Programming*, 27(3):131–159, June 1999.

[72] W. Pugh. The omega test: a fast and practical integer programming algorithm for dependence analysis. *Comm. of the ACM*, 8:102–114, August 1992.

[73] W. Pugh. Counting solutions to presburger formulas: how and why. In *Proceedings of the ACM SIGPLAN 1994 conference on Programming language design and implementation*, pages 121–134. ACM Press, 1994.

[74] P. Quinton, S. V. Rajopadhye, and T. Risset. On manipulating z-polyhedra using a canonical representation. *Parallel Processing Letter*, 7(2):181–194, 1997.

[75] R. Sakellariou. *On the Quest for Perfect Load Balance in Loop-Based Parallel Computations*. PhD thesis, University of Manchester, October 1996.

[76] A. Schrijver. *Theroy of Linear and Integer Programming.* John Wiley and Sons, 1986.

[77] R. Seghir, S. Verdoolaege, K. Beyls, and V. Loechner. Analytical computation of ehrhart polynomials and its application in compile-time generated cache hints. In *International Conference on Compilers, Architecture and Synthesis for Embedded Systems, CASES 2004*, September 2004.

[78] V. Shoup. *A computational introduction to Number Theory and Algebra*. unpublished, July 2004. beta version 4.

[79] R. P. Stanley. Decompositions of rational convex polytopes. *Annals of Discrete Math.*, 6:333–342, 1980.

[80] H.S. Stone and D.Thiebaut. Footprints in the cache. In *Proc. ACM SIGMetrics*, pages 4–8, May 1986.

[81] N. Tawbi. Estimation of nested loop exectution time by integer arithmetics in convex polyhedra. In *Proc. of the 1994 International Parallel Processing Symposium*, April 1994.

[82] N. Tawbi and P. Feautrier. Processor allocation and loop scheduling on multiprocessor computers. In *Proc. of the 1992 International Conference on Supercomputing*, pages 63–71, July 1992.

[83] S. Verdoolaege, K. Beyls, M. Bruynooghe, R. Seghir, and V. Loechner. Analytical computation of ehrhart polynomials and its applications for embedded systems. In *Workshop on Optimizations for DSP and Embedded Systems, in conjunction with IEEE/ACM International Symposium on Code Generation*, March 2004.

[84] D.K. Wilde. A library for doing polyhedral operations. Technical report 785, IRISA, Rennes, France, 1993.

[85] 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.

[86] M. Wolfe. *High Performance Compilers for Parallel Computing.* Addison Wesley, 1996.

[87] J. Xue. Unimodular transformations of non-perfectly nested loops. *Parallel Computing*, 22(12):1621–1645, 1997.

Personal bibliography

Refereed International Journals

- V. Loechner, B. Meister and Ph. Clauss, Precise data locality optimization of nested loops, *The Journal of Supercomputing*, January 2002, volume 21(1), pages 37-76, Kluwer Academic Pub.

- Ph. Clauss and B. Meister, Automatic Memory Transformations to Optimize Spatial Locality in Parameterized Loop Nests, *ACM SIGARCH, Computer Architecture News*, March 2000, volume 28(1), pages 11-19.

Refereed International Conferences

- B. Meister, Periodic Polyhedra, *13th International Conference on Compiler Construction, CC 2004, Part of ETAPS 2004*, Evelyn Duesterwald *Editor*, pages 134-149, LNCS 2985, Springer. April 2004, Barcelona, Spain.

- V. Loechner, B. Meister and Philippe Clauss, Data sequence Locality: a Generalization of Temporal Locality, *EuroPar, European Conference on Parallel Computing*, LNCS 2150, Springer. August 2001, Manchester, United Kingdom.

- Ph. Clauss and B. Meister Automatic Memory Layout Transformation to Optimize Spatial Locality in Parameterized Loop Nests, *4th Annual Workshop on Interaction between Compilers and Computer Architectures INTERACT-4*. February 2000, Toulouse, France.

International Conferences

- B. Meister, Projecting Periodic Polyhedra for Loop Nest Analysis, *CPC 2004, 11th Workshop on Compilers for Parallel Computers.* July 2004, Chiemsee, Germany.
- Ph. Clauss, V. Loechner and B. Meister, Minimizing Strides in Loops with Affine Array References, *CPC 2001, 9th Workshop on Compilers for Parallel Computers.* June 2001, Edinburgh, United Kingdom.

Refereed Francophonic Conferences

- B. Meister, Localité de données dans les opérations stencil, *13es Rencontres francophones du parallélisme, RenPar'13*, pages 37-42. April 2001, Paris, France.

Francophonic Conferences

- B. Meister, Une nouvelle méthode de calcul de l'enveloppe entière d'un polyèdre paramétré, *JPOC 2003, Journée Polyèdres and Optimisation Combinatoire*, LIMOS. June 2003, Clermont-Ferrand, France.

Research reports

- B. Meister, Using Periodics in Integer Polyhedral Problems, Research report of the ICPS-LSIIT, 2003, http://icps.u-strasbg.fr/~meister.
- B. Meister, Localité de données et compression mémoire dans les opérations stencil, Research report of the ICPS-LSIIT, September 2001, http://icps.u-strasbg.fr/~meister.
- B. Meister, V. Loechner and Ph. Clauss, The Polytope Model for Optimizing Cache Locality, Research report of the ICPS-LSIIT, May 2000, http://icps.u-strasbg.fr/