Deriving Formulae to Count Solutions to Parameterized Linear Systems using Ehrhart Polynomials: Applications to the Analysis of Nested-Loop Programs

Philippe Clauss †, Vincent Loechner †, Doran Wilde ‡

† ICPS, Université Louis Pasteur de Strasbourg,
Pôle API, Boulevard Sébastien Brant, 67400 Illkirch, France
e-mail : {clauss,loechner}@icps.u-strasbg.fr

‡ Brigham Young University
Department of Electrical and Computer Engineering
459 Clyde Building, Provo, Utah 84602
e-mail : wilde@ee.byu.edu

April 10, 1997

Keywords: Polyhedron, Enumeration, Parametric Analysis, Ehrhart Polynomial, Symbolic Analysis, Nested-Loop Programs, Lattice, Parallelizing Compiler, Volume

Abstract

Optimizing parallel compilers need to be able to analyze nested loop programs with parametric affine loop bounds, in order to derive efficient parallel programs. The iteration spaces of nested loop programs can be modeled by polyhedra and systems of linear constraints. Using this model, important program analyses such as computing the number of flops executed by a loop, computing the number of memory locations or cache lines touched by a loop, and computing the amount of processor to processor communication needed during the execution of a loop— all reduce to the same mathematical problem: finding the formula for number of integer solutions to a system of parameterized linear constraints, as a function of the parameters.

In this paper, we present a method for deriving a closed-form symbolic formula for the number of integer points contained in a polyhedral region in terms of size parameters. If the set of integer points to be counted lies inside a union of rational convex polytopes, then the number of points can be formulated by a special kind of polynomial called an *Ehrhart pseudo-polynomial*. The procedure consists of first, computing the *parametric vertices* of a polytope defined by a set of parametric linear constraints, and then solving for the pseudo-polynomial which gives the exact formula for the number of integer points in the polytope. An algorithm to derive this formula is

presented. The method is then illustrated using many examples from the literature and in the area of parallel program optimization.

1 Introduction

Counting the number of integer solutions to a system of linear constraints, and counting the number of integer points in a polytope are duals of the same problem. If the constraints are expressed in terms of one or more parameters (the polytope has vertices expressed in terms of one or more parameters), then the count can be expressed by a formula which is a function of those parameters. This paper shows how such a formula may be derived.

Kuck [15] showed that the iteration domain of a nested-loop program with affine lower and upper bounds can be described in terms of a polyhedron. The *polyhedral model* has since been used to model and analyze static nested-loop programs. As a result, fundamental mathematical problems in the areas of discrete and convex geometry, enumerative combinatorics, geometry of numbers and linear programming have been posed in connection with the analysis of nested-loop programs. One of these fundamental problems is finding a formula for the number of integer points contained in a parameterized polytope. Moreover, the formulation needs to be parameterized, since scientific algorithms (and hence the polyhedra that model them) often depend on size parameters.

Such an enumeration formula is helpful when analyzing parallel algorithms. In order to produce efficient parallel programs, a compiler may need to know the number of operations performed by a loop, the number of memory locations or cache lines touched by a loop, or the number of array elements that need to be transmitted from a processor to another during the execution of a loop. Such analyses pose the same general problem: counting the number of integer solutions of selected free variables in a set of linear constraints. The resulting information can be used to estimate the execution time of a code segment, compare the memory bandwidth requirements versus the flop count of a code segment [21], determine which loops will flush the cache, and then calculate the cache miss rate [23, 11], evaluate message traffic and allocate message buffers, determine whether a parallel loop is load balanced (i.e., performs the same amount of computation in each iteration [27]), or if not balanced, determine number of iterations to assign to each processor in order to balance the work load (balanced chunk-scheduling) [13], exploit maximum parallelism, optimize the number of needed processors, and optimize loop and data partitioning.

A *parameterized* linear system is specified by a set of parametric linear constraints which linearly depend on one or more positive integral size parameters $n_1, n_2, \cdots$. A parameterized linear constraint can be written as:

$$\begin{aligned} \sum_i a_i x_i &\leq \sum_j b_j n_j + c \qquad \text{or,} \\ \sum_i a_i x_i &= \sum_j b_j n_j + c \end{aligned}$$

where the a_i's, b_i's and c are rational constants, the n_j's are the size parameters and the x_i's are free variables. A system of such constraints defines a lattice of integral points $x \in \mathcal{Z}^k$ contained in a convex polytope. We want to know the number of lattice points contained in a parameterized polytope, or in other words, the number of integer solutions to the parameterized system of constraints.

This number can be expressed as a symbolic sum, which is not generally an easy formulation. Some symbolic mathematical packages such as Maple or Mathematica can generate such symbolic sums, however they assume that the lower bound is never greater than the upper bound. If this assumption is violated, then their answer is wrong. For example, Mathematica computes that

$$\sum_{i=1}^{n}\sum_{j=i}^{m} 1 = \frac{n(2m-n+1)}{2}$$

This answer is valid only if $n \leq m$. A different formula, $\frac{m(m+1)}{2}$, is required for the case $n > m$. In general, the symbolic number of integer solutions to a system of constraints may differ for different subdomains of the size parameters. This fact was pointed out by the French mathematician Eugène Ehrhart in 1977 [10, p. 139] based on work he did in the late 1950's:

Conjecture 1 (Ehrhart's conjecture) *For any significant diophantine linear system of any dimension, linearly dependent on several positive integral parameters, the symbolic number of integer solutions as a function of those parameters is expressed at different subdomains of the parameter space by different pseudo-polynomials.*

This conjecture is proved later in this paper (theorem 2). A *pseudo-polynomial* is a polynomial where the values of the coefficients rotate cyclically through an array of constants according to the modulos of the parameters. A more formal definition is given in the next section. Ehrhart showed that the values of these coefficients are closely related to the vertices of the parameterized polytope. The validity domains associated with the pseudo-polynomials are also associated with the parametric coordinates of the vertices.

Earlier methods to compute the coefficients of a pseudo-polynomial were proposed using several approaches. For example, Barvinok[1] [2] proposed a complexity study of the computation of these coefficients in terms of computation of the volumes of faces, since any coefficient can be related to volumes or surface areas of an associated polyhedron. But since the problem of volume computation is NP-hard [6], results given by Barvinok relied on the strong assumption of a *Volume Computation Oracle*.

In this paper, we present a new algorithmic method for computing the formula for the number of integer points contained in a parameterized polytope, that is, a polytope defined by a set of linear constraints depending linearly on a set of parameters. The first step of this method consists of computing the parametric vertices of the polytope. To find the vertices of parameterized polyhedra, a new algorithm developed by Loechner and Wilde [17] is used. This algorithm partitions the parameter domain into subdomains where the vertices of a parametric polyhedron are stable, and it provides the information about the vertices that is needed to apply Ehrhart's results. We have extended Ehrhart's results that were based on a single positive integral parameter, to any number of positive integral parameters. The implementation of this extension was made possible by the use of the Loechner-Wilde algorithm. Using this information, a pseudo-polynomial is formulated with unknown coefficients that meets Ehrhart's criteria for being an enumerator for the polytope. Then, a certain number of instances of the polytope are scanned for a range of small fixed values of the parameters and the number of interior points in each instance is counted. Enough information is gathered to build

[1] Barvinok also proposed in [1] a polynomial-time algorithm for counting integral points in non-parametric polyhedra, i.e. when the values of the parameters are fixed.

a system of linear equations to solve for the unknown coefficients. Finally, the pseudo-polynomial is symbolically simplified according a set of rewrite rules. This procedure is repeated for each of the parameter subdomains identified by the Loechner-Wilde algorithm. The final result is a piecewise pseudo-polynomial enumerator formula for the polytope.

This paper will be presented as follows. In section 2, we begin by presenting the *polyhedral model* and by giving some geometric and arithmetic definitions and some of Ehrhart's contributions. Then, in section 3 we present the definition and properties of *pseudo-polynomials* and show how they are used to formulate the number of lattice points inside a union of convex polytopes. In section 4, we describe the algorithm used to derive these formula. In section 5, the method presented here is illustrated with many examples and compared with related works [11, 27, 13, 14, 26, 21]. Many applications can make good use of a symbolic formula for the number of integral solutions to system of parameterized constraints. In section 6, we will show example applications in which the results in this paper can be effectively used for the analysis and transformation of scientific programs. In section 7 we briefly suggest to how our ideas may be applied to certain non-linear systems. And finally, in section 8, we summarize what we have presented.

2 The polyhedral model

We first recall some basic notions dealing with geometry of numbers [12, 22, 3] and enumerative combinatorics [24, 25]. Then some more specific concepts, closely related to the content of the paper, are introduced.

2.1 Polyhedra and polytopes

Let $\mathcal{Q}$ denote the set of rational numbers and $\mathcal{Z}$ the set of integers. A *(convex) polyhedron* is defined by a finite system of linear inequalities $P = \{x \in \mathcal{Q}^d \mid Ax \leq b\}$ where A is a rational matrix and b a rational vector. A bounded convex polyhedron is called a convex *polytope*, and represents the convex hull of a finite number of points $x_1, \ldots, x_{m+1}$ in $\mathcal{Q}^d$. If $x_1, \ldots, x_{m+1}$ are affinely independent points of $\mathcal{Q}^d$ (which also means that $x_2 - x_1, x_3 - x_1, \ldots, x_{m+1} - x_1$ are linearly independent) then the convex hull of $x_1, \ldots, x_{m+1}$ is called an *m-dimensional simplex* of $\mathcal{Q}^d$ with vertices $x_1, \ldots, x_{m+1}$.

Since our goal is to count the number of integer points contained in a polyhedron, it must be bounded, and thus only polytopes are considered in the rest of the paper.

The *affine span* of P is the smallest affine subspace of $\mathcal{Q}^d$ entirely containing P. The *dimension* of P is the dimension of its affine span, and a k-dimensional polytope is called a k-polytope for brevity.

If $\vec{c}$ is a nonzero vector and γ is a rational constant defined by $\gamma = MAX\{\vec{c}x \mid x \in P\}$, the affine hyperplane $\{x \mid \vec{c}x = \gamma\}$ is called a *supporting hyperplane* of P. The *faces* of P are $\emptyset$, P, and the intersections of P with its supporting hyperplanes. Each face of P is itself a polyhedron and is called a k-face if it is k-dimensional. If P is a k-polyhedron, faces of P of dimension 0, 1 and $k-1$ are called *vertices, edges* and *facets*, respectively.

2.2 Parameterized lattice polytopes

We consider parameterized polytopes of the form:

$$P_N = \{x \in \mathcal{Q}^d \mid Ax \leq BN + b; Cx = DN + d\}$$

where N is the symbolic p-vector of the parameters, A, B, C, and D are constant rational matrices, and b and d are constant rational vectors. P_N can also be thought of as a family of polytopes where each valid assignment of values to the vector N gives one member of the set.

Parameterized polytopes may also be represented in the *combined* data and parameter space as a polyhedron P' defined as:

$$P' = \left\{ \begin{pmatrix} x \\ N \end{pmatrix} \in \mathcal{Q}^{d+p} \mid [A \mid -B] \begin{pmatrix} x \\ N \end{pmatrix} \leq c; [C \mid -D] \begin{pmatrix} x \\ N \end{pmatrix} = d \right\}$$

2.3 Lattices

A *lattice* is the set of points $\mathcal{L}^d = \{\alpha_1 \vec{g}_1 + ... + \alpha_d \vec{g}_d \mid \alpha_1, \cdots, \alpha_d \in \mathcal{Z}\}$ where $\vec{g}_1, ..., \vec{g}_d$ are linearly independent vectors of $\mathcal{Q}^d$, called the *basis* of the lattice. The set of all the integral points is called *standard lattice* denoted by $\mathcal{Z}^d$. Let G be the matrix whose columns are the vectors $\vec{g}_1, ..., \vec{g}_d$, called the *dilatation matrix*. Any lattice $\mathcal{L}^d$ consists of the points $\mathcal{L}^d = \left\{x \mid x = Gz, z \in \mathcal{Z}^d\right\}$. A *lattice polytope* or a *lattice simplex* is then defined by a set of linear inequalities of the form $\{z \in \mathcal{L}^d \mid Az \leq b\}$ where A is a $m \times d$ rational matrix and b a m-dimensional rational vector. The intersection between a lattice $\mathcal{L}^d$ and a polytope is called a *lattice polytope* P of $\mathcal{L}^d$. A lattice polytope P of $\mathcal{L}^d$ is a $\mathcal{L}^d$-polytope if all of its vertices are in $\mathcal{L}^d$. If $\mathcal{L}^d = \mathcal{Z}^d$, P is an *integral polytope*. If some vertices of P do not belong to $\mathcal{L}^d$ and have rational coordinates, P is a *rational* polytope. The *enumeration* of any set E is the number of points in $E \cap \mathcal{L}^d$. An *enumerator* is a formula to give the number of points in $E \cap \mathcal{L}^d$.

Since by definition, any lattice $\mathcal{L}^d$ can be derived from $\mathcal{Z}^d$ by a non-singular transformation G [12], any $\mathcal{L}^d$-polytope is affinely equivalent to some integral polytope. Hence, results given in the rest of this paper concerning integral polytopes can be directly extended to $\mathcal{L}^d$-polytopes. From now on and without loss of generality, the generic term of integral polytopes will be used.

2.4 Ehrhart polyhedra

Many scientific algorithms are defined parametrically in order to apply to any size of problem. Implementations of these algorithms yield programs with loops involving parameterized affine loop bounds. When modeling such loops with convex polytopes, parameterized polytopes have to be considered. Eugène Ehrhart defined a type of parameterized polytope which he called *homothetic* and *bordered systems* [8, 9] that are similar to what was defined above in section 2.2. However, his definitions were limited to homothetic and bordered systems of only a single parameter n. We will extend his results to the general case of any number of parameters.

In this work, we also consider polytopes which may be non-convex, called *normal polyhedra*, according to the following definition. This definition corresponds to the one given by Ehrhart in [8] and [10, p. 47] and includes unions of convex polytopes.

Definition 1 (normal polyhedron)
A **normal polyhedron** *P is a closed, not necessarily convex polyhedron which admits a finite rectilinear simplicial subdivision. It is defined to be the union of a finite number of simplices $S_1, S_2, ..., S_n$ of equal dimension such that,*

- *The intersection between any two simplices is either empty or is a common face,*

- *Each facet of one of these simplices belongs to at most one other simplex. Faces belonging to only one simplex form the border of the polyhedron,*
- *Assuming $P_h = S_1 \cup S_2 \cup ... \cup S_h$, these simplices can be ordered such that for any $h < n$, the intersection $P_h \cap S_{h+1}$ is either empty, or is the union of facets common to S_{h+1} and the border of P_h.*

Our method of counting the number of integer points is based on the decomposition of a parametric polyhedron into several *homothetic and bordered* systems, associated with *validity domains*, according to the following definition which is an extension of Ehrhart's definition [8] to any number of parameters.

Definition 2 (homothetic system)
A system H_N, $N = (n_1, \cdots, n_q)$ consists of a set of **homogeneous** *constraints of the form $\sum a_i x_i = \sum b_j n_j$, $\sum a_i x_i < \sum b_j n_j$, $\sum a_i x_i \leq \sum b_j n_j$, where the a_i's and the b_j's are integer constants, the x_i's are free variables and the n_j's are positive integral parameters. H_N defines a (convex) polyhedron P_N.*

Such a system is **significant** *if and only if $H_{(1,1,...,1)}$ defines a (convex) polytope $P_{(1,1,...,1)}$.*

H_N is **homothetic** *if all the q one-parameter polytopes $P_{0,0,...,0,n_j,0,...,0}$, $1 \leq j \leq q$, each correspond to $P_{0,0,...,0,1,0,...,0}$ by a n_j-homothetic transformation [2] of the origin O. All the $P_{0,0,...,0,1,0,...,0}$'s are called the* **primitive domains** *of H_N.*

The properties of such homothetic polyhedra yield the main mathematical result given later in the paper. This definition can be extended to *bordered polytopes* in order to handle polytopes modeling loops with *affine* loop bounds. The following definition is also an extension of Ehrhart's [8].

Definition 3 (bordered system)
A system B_N, $N = (n_1, \cdots, n_q)$ consists of a set of **affine** *constraints of the form $\sum a_i x_i = \sum b_j n_j + c$, $\sum a_i x_i < \sum b_j n_j + c$, $\sum a_i x_i \leq \sum b_j n_j + c$, where the a_i's, b_j's, and c are integer constants, the x_i's are free variables and the n_j's are positive integral parameters. B_N defines a (convex) polyhedron P'_N.*

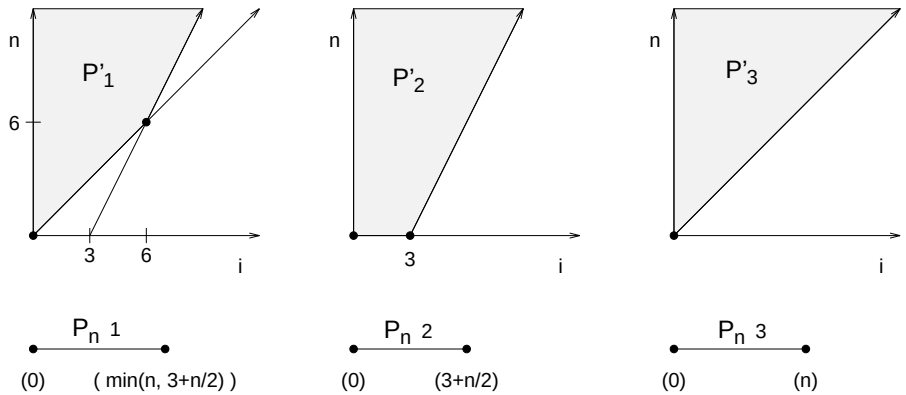
The system B_N is **bordered** *if removing all the c's results in a homothetic system H_N which defines a polyhedron P_N. P'_N may be obtained by intersecting hyperplanes parallel to each facet of P_N at a rational distance from the facet independent of the parameters N. P_N is called the* **kernel** *of P'_N.*

Observe that the borders can cut the kernel while the parameters are less than certain minimal values. Hence, P'_N is said to be *complete* if it exists, i.e. when all the borders are involved as bounds. Observe also that a translation, independent of the parameters, transforms an homothetic polytope P_N into a bordered polytope, complete for all values of the parameters.

In general, a parameterized polyhedron is a set of homothetic or bordered polyhedra qualified by subdomains of the parameter space called *validity domains*. These subdomains are computed by the algorithm that finds the parametric vertices described in a later section.

[2] Let O be a point in a vector-space $\mathcal{R}^d$, P be a set of points in $\mathcal{R}^d$, and k be a real number not equal to zero. Then, the *k-homothetic transformation* of P about a center point O is defined as $\{ N \in \mathcal{R}^d \mid M \in P;\ \overrightarrow{ON} = k\ \overrightarrow{OM} \}$.

Example 1



In the above figure, we have three parameterized, one dimensional polyhedra,

$$\begin{aligned} P_{n1} &= \{i \mid i \geq 0; i \leq n; 2i \leq n+6\} \\ P_{n2} &= \{i \mid i \geq 0; n \geq 0; 2i \leq n+6\} \\ P_{n3} &= \{i \mid 0 \leq i \leq n;\} \end{aligned}$$

P_{n3} is a homothetic system, and P_{n2} is a bordered system. P_{n1} is neither homothetic nor bordered, but can be made as a union of homothetic and bordered systems qualified by certain subdomains of the parameter (n) space,

$$P_{n1} = \cup \left\{ \begin{array}{ll} P_{n2} & \text{when} \quad 0 \leq n \leq 6 \\ P_{n3} & \text{when} \quad 6 \leq n \end{array} \right.$$

□

The results of this paper apply to any k-polyhedron P_N that is a union of homothetic or bordered systems of the form $\{z \in \mathcal{Q}^d \mid Az \leq BN + b; Cz = DN + d\}$, where $N = (n_1, \ldots, n_q)$.

2.5 Pseudo-polynomials

The enumerator of a parameterized polyhedron can be formulated using *polynomials* and *pseudo-polynomials* according to the following definitions. Before we can define pseudo-polynomials, we must first introduce the concept of periodic numbers. We extend Ehrhart's definition of 1-dimensional periodic numbers [10] to any dimension.

Definition 4 (periodic number) [3]
A one-dimensional periodic number $u(n) = [u_0, u_1, \ldots, u_{p-1}]_n$ *is equal to the item whose index is equal to* $(n \bmod p)$. p *is called the* **period** *of* $u(n)$.

$$u(n) = [u_0, u_1, \ldots, u_{p-1}]_n = \left\{ \begin{array}{ll} u_0 & \textit{if } (n \bmod p) = 0 \\ u_1 & \textit{if } (n \bmod p) = 1 \\ \vdots & \vdots \\ u_{p-1} & \textit{if } (n \bmod p) = p-1 \end{array} \right\} = u_{(n \bmod p)}$$

[3] Ehrhart originally defined a periodic number to be $u(n) = [u_1, u_2, \ldots, u_p]_n = \left\{ \begin{array}{ll} u_1 & \text{if } (n \bmod p) = 1 \\ u_2 & \text{if } (n \bmod p) = 2 \\ \ldots & \\ u_p & \text{if } (n \bmod p) = 0 \end{array} \right.$
The definition used in this paper has been modernized to index the array u starting at 0, instead of 1.

Let $N = (n_1, n_2, \cdots, n_q)$ be a q-dimensional parameter vector. A q-dimensional periodic number $u(N)$ is defined by a q-dimensional array U of size $p_1 \times p_2 \times \ldots \times p_q$ such that

$$u(N) = u_{n_1,n_2,\cdots,n_q} = U_{(n_1 \bmod p_1, n_2 \bmod p_2, \cdots, n_q \bmod p_q)}$$

q is the **dimension** *of $u(N)$. The vector $p = (p_1, p_2, \cdots, p_q)$ is called the* **multi-period** *of $u(N)$ and the least common multiple of the p_i's is called the* **period** *of $u(N)$.*

Example 2

$(-1)^n = [1, -1]_n$

$$(-1)^{n-m} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}_{(n,m)} = [\ [1,-1]_m, [-1,1]_m]_n$$

Note that the matrix array notation works well for 2-dimensional periodic numbers, but the vector of vectors notation works for all dimensions.

□

Ehrhart [10] gave the basic properties of one-dimensional periodic numbers which can easily be extended to the multi-dimensional case:
Let $u(N)$ and $u'(N)$ be q-periodic numbers of multi-period p and p' respectively. They can be reduced to the same multi-period $m = (m_i)$, where $m_i = p_i\alpha = p'_i\beta$ is the least common multiple of p_i and p'_i, by repeating α times (respectively β times) the sequence of items in the i^{th} dimension of $u(N)$ (respectively $u'(N)$). Since $u(N)$ and $u'(N)$ are reduced to the same period, $u(N)+u'(N) = (U_I+U'_I)_N$ and $u(N)u'(N) = (U_I U'_I)_N$, where $I = (n_1 \bmod p_1, n_2 \bmod p_2, \cdots, n_q \bmod p_q)$.

Ehrhart defined pseudo-polynomials for a single variable n and 1-dimensional periodic coefficients.

Definition 5 (pseudo-polynomial)
A polynomial

$$f(n) = c_k n^k + c_{k-1} n^{k-1} + \cdots + c_0$$

is called a (1-dimensional) **pseudo-polynomial** *of degree k if its coefficients c_i are 1-dimensional periodic numbers, and $f(n)$ is integral for all n. We have extended this definition to cover multi-dimensional polynomials.*

Let $N = (n_1, n_2, \cdots, n_q)$ be a q-dimensional parameter vector. A q-dimensional pseudo-polynomial $f(N)$ of degree k is defined as

$$f(N) = \sum_{i_1=0}^{k} \sum_{i_2=0}^{k} \cdots \sum_{i_q=0}^{k} c_{i_1,i_2,\ldots,i_q} n_1^{i_1} n_2^{i_2} \ldots n_q^{i_q}$$

where the coefficients $c_{i_1,i_2,\ldots,i_q}$'s are periodic numbers whose dimensions are at most q and $f(N)$ is integral for all N.

A pseudo-polynomial has three characteristics: its dimension, degree, and its pseudo-period. The **dimension** *of the polynomial $F(N)$ is the number of parameter variables q, the* **degree** *of the $F(N)$ is the greatest exponent k, and the* **pseudo-period** *of $F(N)$ is the least common multiple of the periods of its coefficients c_i.*

Example 3

The summation $\sum_{i=0}^{n/2}\sum_{j=0}^{m/2}(1)$ can be formulated as a pseudo-polynomial of 2 variables, n and m, of degree 2, and period 2. $\begin{aligned}\sum_{i=0}^{n/2}\sum_{j=0}^{m/2}(1) &= \left(\tfrac{1}{4}m+\left[\tfrac{1}{2},\tfrac{1}{4}\right]_m\right)n+\left[\left(\tfrac{1}{2}m+\left[1,\tfrac{1}{2}\right]_m\right),\left(\tfrac{1}{4}m+\left[\tfrac{1}{2},\tfrac{1}{4}\right]_m\right)\right]_n \\ &= \tfrac{1}{4}nm+\left[\tfrac{1}{2},\tfrac{1}{4}\right]_m n+\left[\tfrac{1}{2},\tfrac{1}{4}\right]_n m+\left[\begin{array}{cc}1 & 1/2\\ 1/2 & 1/4\end{array}\right]_{n,m}\end{aligned}$

□

3 Parametric enumerators with pseudo-polynomials

In this section, we recall and extend some results due to Ehrhart who made a study of the enumeration of lattice points in polyhedra by means of polynomials along the same lines as Euler. All the initial proofs can be found in [7, 8, 9, 10]. Some aspects of Ehrhart's work were corrected, streamlined, and expanded by I. G. Macdonald [18, 19] and R. P. Stanley [24, 25].

The **enumerator** of a normal polyhedral domain is the number of integer lattice points contained in the domain. We denote the enumerator of a parameterized polyhedron P_N as E_N.

Definition 6 (denominator of a polyhedron)
The **denominator of a rational point** *is the least common multiple of the denominators of its coordinates.*
The **denominator of a rational polyhedron** *is the least common multiple of the denominators of its vertices.*

Example 4

The denominator of a point $\left(\frac{1}{2},\frac{5}{3}\right)$ is $lcm(2,3)=6$.

The denominator of a polyhedron with vertices: $\left(\frac{1}{2},\frac{5}{3}\right)$, $\left(\frac{3}{2},\frac{9}{4}\right)$, and $\left(-\frac{1}{2},\frac{1}{6}\right)$ is $lcm(6,4,6)=12$

□

Ehrhart's main result consists in the following theorem dealing with parametric polyhedra which depend on a single parameter n.

Theorem 1 (Ehrhart's fundamental theorem)
The enumerator E_n of any homothetic or bordered k-polytope P_n is a polynomial in n of degree k if P_n is integral, and it is a pseudo-polynomial in n of degree k whose pseudo-period is the denominator of P_n if P_n is rational.

We extend this theorem to any number of parameters $N=(n_1,\ldots,n_p)$.

Theorem 2 (extended fundamental theorem)
Given any homothetic or bordered d-polytope $\mathcal{P}_N$ defined in terms of p linearly-independent parameters and having a denominator q, the enumerator E_N of such a polytope can be expressed as a p-variable pseudo-polynomial:

$$E(N)=\sum_{i_1=0}^{d}\sum_{i_2=0}^{d-i_1}\cdots\sum_{i_p=0}^{d-i_1-i_2-\cdots-i_{p-1}} c_{i_1,i_2,\ldots,i_p}\,n_1^{i_1}n_2^{i_2}\ldots n_p^{i_p}$$

whose terms have degree at most d, and whose the coefficients $c_{i_1,i_2,\ldots,i_p}$'s are periodic numbers with periods at most q.

Inasmuch as any polytope is composed of different homothetic or bordered polytopes over different subdomains of the parameters, the enumerator of any polytope can be expressed as different piecewise pseudo-polynomials defined over those same parameter subdomains.

The proof uses the following lemma stating that if a parametric polytope is defined by affine constraints, then the coordinates of its vertices are also affine functions of the parameters.

Lemma 1 (parametric vertices)
If P_N is a parametric polytope defined by affine constraints of the form $ax \leq bN + c$, then the parametric coordinates of its vertices are also affine functions of the parameters of the form $dN + e$, where e is a rational constant p-vector and e is a rational scalar constant.

Proof (Lemma 1)
Without loss of generality, let P_N be a k-polytope defined by a non-redundant system of constraints. Each vertex is a k-vector that can be defined as the intersection of at least k constraints which are saturated at the vertex. An inequality $Ax \leq BN + c$ is *saturated* when the equality $Ax = BN + c$ is true. The resolution of such a system of equations derived from k saturated constraints leads to a solution where each element of the x vector is expressed as an affine function of the parameters. We note that if the constant c is zero for all constraints, that all constraints are linear functions of N and all coordinates of vertices are likewise linear functions of N. Such a system is *homothetic*. Systems with non-zero c's are *bordered* and have affine constraint systems and affine coordinate vertices.
□

We note with interest that the reverse of lemma 1 is *not* always true. The convex hull of a set of parameterized vertices in a data space does not always generate a corresponding polyhedron in the combined data and parameter space. Here is a counter-example.

Example 5

Consider a line segment in 2-space with end points at $[1, 1]$ and $[2, n]$, where n is a positive parameter. Then a point on the closed segment connecting $[1, 1]$ and $[2, n]$ can be written as $[i, j] = [1 + k(2 - 1), 1 + k(n - 1)]$, for $0 \leq k \leq 1$. Solving for k: $k = i - 1 = \frac{j-1}{n-1}$ leading to $(j - 1) = (i - 1)(n - 1)$ which shows that i, j, and n are not *linearly* related, and thus the set of points in i, j, n-space corresponding to points on the parametric closed segment between $[1, 1]$ and $[2, n]$, is not a polyhedron. [Example due to Sanjay Rajopadhye.]
□

Proof (Theorem 2)
Part 1

We use an inductive proof to show that the enumerator of P_N is a pseudo-polynomial. We use as the induction variable, p, the number of parameters.
Base Case: Consider a bordered polytope P_n depending on one parameter n, $(p = 1)$. The enumerator of P_n is a pseudo-polynomial by application of theorem 1.
Inductive Step: Assume that the enumerator of a bordered polytope $P_{n_1,\ldots,n_{p-1}}$ depending on $p - 1$ parameters is a pseudo-polynomial.

Now, consider a parametric polytope $P_N = P_{n_1,\ldots,n_p}$ depending on p parameters. By fixing parameters $n_1, \ldots, n_{p-1}$ to constants, and leaving only n_p a free parameter, we obtain a bordered polytope $P_{n_p}(n_1, \ldots, n_{p-1})$. By application of theorem 1, the enumerator of $P_{n_p}(n_1, \ldots, n_{p-1})$ can be written:

$$E_{n_p}(n_1, \ldots, n_{p-1}) = c_d(n_1, \ldots, n_{p-1})n_p^d + c_{d-1}(n_1, \ldots, n_{p-1})n_p^{d-1} + \ldots + c_0(n_1, \ldots, n_{p-1})$$

By substituting n_p in the above equation with some initial values 1, 2, ... we get a system of linear equations of the form:

$$\begin{cases} c_d(n_1, \ldots, n_{p-1})_1 1^d + c_{d-1}(n_1, \ldots, n_{p-1})_1 1^{d-1} + \ldots + c_0(n_1, \ldots, n_{p-1})_1 & = & E_{n_1,\ldots,n_{p-1},1} \\ c_d(n_1, \ldots, n_{p-1})_2 2^d + c_{d-1}(n_1, \ldots, n_{p-1})_2 2^{d-1} + \ldots + c_0(n_1, \ldots, n_{p-1})_2 & = & E_{n_1,\ldots,n_{p-1},2} \\ \vdots & & \vdots \end{cases}$$

The enumerators $E_{n_1,\ldots,n_{p-1},1}, E_{n_1,\ldots,n_{p-1},2}, \ldots$ are pseudo-polynomial by the inductive assumption. Enough initial values for n_p can be chosen so that a solution for the coefficients $c_i(n_1, \ldots, n_{p-1}), d \geq i \geq 0$ can be derived. Each coefficient $c_i(n_1, \ldots, n_{p-1})$ will be in fact a linear combination of the pseudo-polynomials $E_{n_1,\ldots,n_{p-1},1}, E_{n_1,\ldots,n_{p-1},2}, \ldots$ and therefore each coefficient $c_i(n_1, \ldots, n_{p-1})$ will itself be a pseudo-polynomial. And since a pseudo-polynomial with pseudo-polynomial coefficients is itself a pseudo-polynomial, E_N is also a pseudo-polynomial. This completes part 1 of the proof.

Part 2

In this part, we show that no term of E_N has a degree greater than d. $\mathcal{P}_N$ has a finite number of vertices $v_k, 1 \leq k \leq r$, whose indices $v_k[i], 1 \leq i \leq d$ are affine functions of the parameters, $n_1, n_2, \cdots, n_p$ (by Lemma 1). The rectangular bounding box of $\mathcal{P}$ has an enumeration of

$$\begin{gathered} (\max_{(1\leq k\leq r)} v_k[1] - \min_{(1\leq k\leq r)} v_k[1] + 1)\times \\ (\max_{(1\leq k\leq r)} v_k[2] - \min_{(1\leq k\leq r)} v_k[2] + 1)\times \\ \vdots \\ (\max_{(1\leq k\leq r)} v_k[d] - \min_{(1\leq k\leq r)} v_k[d] + 1) \end{gathered}$$

Since each term $v_k[i]$ is an affine function of the parameters, the term $\max_{(1\leq k\leq r)} v_k[d] - \min_{(1\leq k\leq r)} v_k[d] + 1$. will also be an affine function of the parameters. The product of d affine functions of the parameters

$$\begin{gathered} (a_{10} + a_{11}n_1 + a_{12}n_2 + \cdots + a_{1p}n_p)\times \\ (a_{20} + a_{21}n_1 + a_{22}n_2 + \cdots + a_{2p}n_p)\times \\ \vdots \\ (a_{d0} + a_{d1}n_1 + a_{d2}n_2 + \cdots + a_{dp}n_p) \end{gathered}$$

is a polynomial

$$\sum_{i_1=0}^{d} \sum_{i_2=0}^{d-i_1} \cdots \sum_{i_p=0}^{d-i_1-i_2-\cdots-i_{p-1}} c_{i_1,i_2,\ldots,i_p} n_1^{i_1} n_2^{i_2} \ldots n_p^{i_p}$$

The actual enumeration of $\mathcal{P}$ is always bounded by the enumeration of the bounding box of $\mathcal{P}$. If the pseudo-polynomial $E(N)$ that was the actual enumerator of $\mathcal{P}$ had a term whose order of complexity exceeded any term in the enumerator $E_{BB}(N)$ of the bounding box of $\mathcal{P}$, that is, if there existed some term $c_{i_1,i_2,\ldots,i_p} n_1^{i_1} n_2^{i_2} \ldots n_p^{i_p}$ in $E(N)$ where $i_1 + i_2 + \cdots + i_p > d$, then $O(E(N)) > O(E_{BB}(N))$ and there would exist some value of $N = (n_1, n_2, \cdots, n_p)$ such that $E(N) > E_{BB}(N)$. This leads to a contradiction, and the theorem follows.

□

Parameterized Polyhedron	Pseudo-Polynomial Enumerator
number of parameters	number of variables (dimension)
dimension	degree
denominator	period of coefficients
parameter subdomain for stable vertices	parameter subdomain for formula

Figure 1: Relationship between a parametric polynomial and its enumerator

Due to these theorems, the form of the parametric enumeration formula is known. The relationship between a parameterized polyhedron P_N and its pseudo-polynomial enumerator is summarized in figure 1. The number of variables in (or dimension of) the pseudo-polynomial is equal to the number of (linearly independent) parameters defining P_N. The period of the pseudo-polynomial is equal to the denominator of the polyhedron P_N and the degree of the pseudo-polynomial is equal to the dimension of the polyhedron. And finally, the parameter subdomains over which different pseudo-polynomial are valid, are related to the subdomains where the vertex set of P_N doesn't change (or is stable). Hence, the derivation of the Ehrhart pseudo-polynomial can be done by counting some initial values of the enumerator and solving systems of symbolic rational linear equations. The number of unknown coefficients is of a pseudo-polynomial is $(degree+1)period$ for a one-dimensional polynomial and $((degree+1)period)^{dimension}$ for a multi-dimensional polyhedron. Since the number of equations of the system must be equal to the number of unknown values, the complexity depends on the number of parameters, the number of validity domains, the degree, and the denominator.

The number of unknowns can be reduced using the following theorem, conjectured by Ehrhart [10] and proved by Stanley [25].

Theorem 3 *Let P_n be a rational k-polytope and let the Ehrhart pseudo-polynomial be*

$$E_n = c_k(n)n^k + c_{k-1}(n)n^{k-1} + \ldots + c_0(n)$$

where $c_0, \ldots, c_k$ are periodic numbers of n. If for some $q \in [0,k]$, the affine span of every q-dimensional face of P_1 contains at least one lattice point, then for any $p \geq q$, $c_p(n)$ is constant, i.e. of period one.

Example 6

Consider the set of linear constraints $\left\{0 \leq i \leq \frac{1}{2}n, 0 \leq j \leq \frac{1}{2}m\right\}$ defining the rational 2-polytope $P_{n,m}$,

$$P_{n,m} = \{\binom{i}{j} \in \mathcal{Z}^{\in} \mid \begin{pmatrix} -1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 1 \end{pmatrix} \binom{i}{j} \leq \begin{pmatrix} 0 & 0 \\ \frac{1}{2} & 0 \\ 0 & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \binom{n}{m}\}$$

$P_{n,m}$ is the closed rectangle whose parametric vertices are $(0,0)$, $(\frac{n}{2},0)$, $(0,\frac{m}{2})$, $(\frac{n}{2},\frac{m}{2})$. The denominator of this polyhedron is 2, it is 2-dimensional and is a function of 2 parameters.

By looking at $P_{1,0}$, we observe that one vertex, $(\frac{1}{2},0)$, is not an integer point. Hence, the coefficient of n^0 is periodic. The other coefficients are constant, since all the edges, and obviously the polytope itself, contain at least one integer point.

By looking at $P_{0,1}$, we observe that one vertex, $(0,\frac{1}{2})$, is not an integer point. Hence, the coefficient of m^0 is periodic. The other coefficients are constant, since all the edges, and obviously the polytope itself, contain at least one integer point.

Moreover, the period of the periodic coefficients is equal to the denominator of $P_{n,m}$, i.e. 2. The pseudo-polynomial $E_{n,m} = c_1 n^2 + c_2 n + [c_3, c_4]$ can thus be determined by solving the following system of rational linear equations:

$$\begin{cases} c_3 & = E_{0,m} \\ c_1 + c_2 + c_4 & = E_{1,m} \\ 4c_1 + 2c_3 + c_3 & = E_{2,m} \\ 9c_1 + 3c_2 + c_4 & = E_{3,m} \end{cases}$$

Before we can solve this system, the values of $E_{0,m}, E_{1,m}, E_{2,m}$ and $E_{3,m}$ must be computed first.

We compute $E_{0,m} = c_5 m^2 + c_6 m + [c_7, c_8]_m$:

$$\begin{cases} c_7 & = E_{0,0} = 1 \\ c_5 + c_6 + c_8 & = E_{0,1} = 1 \\ 4c_5 + 2c_6 + c_7 & = E_{0,2} = 2 \\ 9c_5 + 3c_6 + c_8 & = E_{0,3} = 2 \end{cases}$$

We obtain $E_{0,m} = 1, 1, 2, 2$ by counting points in $P_{0,0}$ to $P_{0,3}$ and then solve for $c_5 \cdots c_8$ to obtain $E_{0,m} = \frac{1}{2}m + [1, \frac{1}{2}]_m$.

In a likewise manner we set up systems of equations for $E_{1,m}$, $E_{2,m}$, and $E_{3,m}$, which when solved, give us $E_{1,m} = \frac{1}{2}m + [1, \frac{1}{2}]_m$, $E_{2,m} = m + [2, 1]_m$, and $E_{3,m} = m + [2, 1]_m$.

Finally, we can symbolically solve the first set of equations to obtain:

$$\begin{aligned} E_{n,m} &= \left(\tfrac{1}{4}m + \left[\tfrac{1}{2}, \tfrac{1}{4}\right]_m\right) n + \left[\tfrac{1}{2}m + \left[1, \tfrac{1}{2}\right]_m,\ \tfrac{1}{4}m + \left[\tfrac{1}{2}, \tfrac{1}{4}\right]_m\right]_n \\ &= \tfrac{1}{4}nm + \left[\tfrac{1}{2}, \tfrac{1}{4}\right]_m n + \left[\tfrac{1}{2}, \tfrac{1}{4}\right]_n m + \begin{bmatrix} 1 & 1/2 \\ 1/2 & 1/4 \end{bmatrix}_{n,m} \end{aligned}$$

□

4 Description of the algorithm

To use Ehrhart's results to specify a pseudo-polynomial, we need four pieces of information:

1. the number of parameters
2. the dimension of the system
3. the denominator of system
4. the validity domain over which the polynomial is valid

The number of parameters and the dimension of the system are immediately available from the problem specification. To find the denominator and validity domains of the system, the vertices of the polyhedron P_n must first be computed. We use the algorithm of Loechner and Wilde [17] to determine the parametric coordinates of the vertices of any convex polytope and their associated definition domains. This algorithm is associated with a software package called the *polyhedral library* [28] which eliminates redundant constraints and performs logical operations on polyhedra.

In some problems, non-convex polytopes might have to be considered. In such a case, the Loechner/Wilde algorithm will have to be extended in order to consider Ehrhart's *normal* polyhedron, consisting of a union of convex polytopes $P_N^1, P_N^2, \ldots$ which can only intersect at their proper faces. This proposed extension consists of the following heuristic procedure:

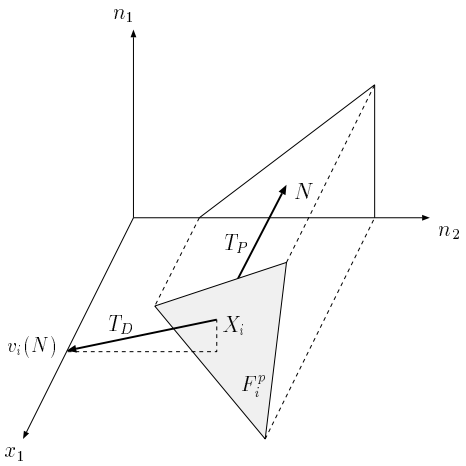


Figure 2: A p-face F_i^p (highlighted) and the computation of $v_i(N)$ as a function of N

- Compute all the vertices of each rational polytope P_N^i.
- Compute all the vertices of each intersection $P_N^i \cap P_N^j, j \neq i$.

4.1 An algorithm to find parametric vertices

The method presented below is fully explained in [17]. It computes the vertices $v_i(N)$ of a parametric polyhedron $P_N = \{x \in \mathcal{Q}^d \mid Ax \leq BN + b;\ Cx = DN + d\}$, where N is a vector of the form $\begin{pmatrix} n_1 \\ \vdots \\ n_p \end{pmatrix}$. If assume that the different n_j are linearly independent, then the following theorem can be shown:

Theorem 4 *Let $v_i(N)$ be a vertex of the polyhedron P_N. Let P' be a polyhedron in the combined data and parameter space corresponding to P_N. Then for each vertex $v_i(N)$ of P_N, there exists a unique corresponding p-face of P' denoted F_i^p such that $\begin{pmatrix} v_i(N) \\ N \end{pmatrix} = F_i^p \cap \left\{ \begin{pmatrix} x \\ n \end{pmatrix} \mid n = N \right\}$.*

Proof The proof can be found in [17]. It is based on the definition of $v_i(N)$ as a 0-face of P_N; $v_i(N)$ is the intersection of a supporting hyperplane $\mathcal{H}_i$ and P_N. In the combined space, $\mathcal{H}_i$ is the intersection of $\left\{ \begin{pmatrix} x \\ n \end{pmatrix} \mid n = N \right\}$ and a supporting hyperplane of P' denoted $\mathcal{H}_i'$. $\mathcal{H}_i'$ can be chosen such that $\mathcal{H}_i' \cap P'$ is a p-face of P' and the result follows.
□

Property 1 *The point $v_i(N)$ is only defined for values of N in the polyhedron $\left\{ N \mid \begin{pmatrix} v_i(N) \\ N \end{pmatrix} \in F_i^p \right\}$, which is a projection of F_i^p onto the parameter space.*

Proof Refer to figure 2. Since $\begin{pmatrix} v_i(N) \\ N \end{pmatrix} = F_i^p \cap \left\{ \begin{pmatrix} x \\ n \end{pmatrix} \mid n = N \right\}$, the validity domain for $v_i(N)$ (the range of N over which $v_i(N)$ exists) is $\left\{ N \mid \begin{pmatrix} v_i(N) \\ N \end{pmatrix} \in F_i^p \right\}$. It is a convex polyhedron since it is a projection of F_i^p and F_i^p is a convex polyhedron.
□

If F_i^p is a p-face of P' corresponding to a vertex $v_i(N)$, then all points $\begin{pmatrix} v_i(N) \\ N \end{pmatrix} \in F_i^p$ lie in the intersection of $d = \text{dimension}(P') - p$ facets of P' and thus must satisfy d linearly independent equations of the form $a^T \begin{pmatrix} v_i(N) \\ N \end{pmatrix} = c$, where a is a constant vector and c is a constant. Using these equations, we can solve for $v_i(N)$ as a vector of linear functions of N.

If such a system of equations cannot be solved, then there does not exist one-to-one correspondence between points in F_i^p and values of N. In this case, the p-face F_i^p not correspond to a vertex of P_N, and that face F_i^p can be ignored by the algorithm.

4.1.1 Partitioning the parameter space

The output of the above algorithm is a set of vertices and for each vertex, a set of constraints that must be verified by the parameters N for that vertex to be valid.

In order to use these results, we need to know the set of distinct parameter subdomains which partition complete *sets* of parameterized vertices, i.e. to transform the collection of (subdomain, vertex) pairs into a collection of (subdomain, set of vertices) pairs, where each subdomain (specified by set of conditions on N) is a partition in which a complete set of vertices of P_N is stable. These partitions are obtained by intersecting the individual domains of all of the vertices in each set.

Ehrhart's theorem (Theorem 1) and our extension to it (Theorem 2) can only be applied to bordered or homothetic polyhedra. These partitions are essential for the application of these theorems because they separate a parameterized polyhedron into a set of bordered and homothetic polyhedra. For each bordered or homothetic polyhedron, a different pseudo-polynomial enumerator is computed.

4.2 Computing periodic coefficients

A pseudo polynomial of degree $p-1$ and period q is written:

$$E_n = [u_{p-1,q-1}, \cdots u_{p-1,0}]_n \, n^{p-1} + [u_{p-2,q-1}, \cdots u_{p-2,0}]_n \, n^{p-2} + \cdots + [u_{0,q-1}, \cdots u_{0,0}]_n \, n^0$$

The values $u_{j,k}$ (where $0 \leq j < p$ and $0 \leq k < q$) are unknowns. To solve for them, we substitute in different values of n to generate different equations.

Normally, a system of pq equations would need to be solved to find pq unknown coefficients taking $\mathcal{O}(p^3 n^3)$ time. However, in this case, we can break the problem down and solve p sets of q-equations, which is done in $\mathcal{O}(pn^3)$ time.

Let $n = n_0$ and $k = (n_0 \bmod q)$. Then substituting n for n_0 in the pseudo-polynomial, we get:

$$u_{p-1,k} n_0^{p-1} + u_{p-2,k} n_0^{p-2} + \cdots + u_{0,k} = E(n_0)$$

If we choose a set of q values for n starting at n_0 and then increasing by strides of q, i.e., $n_i = n_0 + iq$, $0 \leq i < q$, and then substitute those values in the pseudo-polynomial, we arrive at the following

system of equations.

$$\begin{aligned} u_{p-1,k}n_0^{p-1} + u_{p-2,k}n_0^{p-2} + \cdots + u_{0,k} &= E(n_0) \\ u_{p-1,k}n_1^{p-1} + u_{p-2,k}n_1^{p-2} + \cdots + u_{0,k} &= E(n_1) \\ &\vdots \\ u_{p-1,k}n_{q-1}^{p-1} + u_{p-2,k}n_{q-1}^{p-2} + \cdots + u_{0,k} &= E(n_{q-1}) \end{aligned}$$

By striding the value of n by q, the index $k = (n \bmod q)$ remains constant and the same set of u_{jk} values are selected from the periodic coefficients for each equation. This system of equations can be put into matrix format $A\vec{u} = \vec{E}$ as follows:

$$\begin{bmatrix} n_0^{p-1} & n_0^{p-2} & \cdots & 1 \\ n_1^{p-1} & n_1^{p-2} & \cdots & 1 \\ & \vdots & & \\ n_{q-1}^{p-1} & n_{q-1}^{p-2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_{p-1,k} \\ u_{p-2,k} \\ \vdots \\ u_{0,k} \end{bmatrix} = \begin{bmatrix} E(n_0) \\ E(n_1) \\ \vdots \\ E(n_{q-1}) \end{bmatrix}$$

Matrix A will always be an integer matrix and the inverse of A will always be a rational matrix. By inverting matrix A, we can solve for vector $\vec{u}$, obtaining $\vec{u} = A^{-1}\vec{E}$. Two cases arise:

1. The $E(n_j)$'s are constant. They are obtained by counting the number of integer points in the polyhedron $P(n_j)$ (n_j being a fixed integer). This is done by scanning the polyhedron using the algorithm described in [16].

2. The $E(n_j)$'s are not constant. This happens when the enumerator E is a function of more than one parameter, $E(n, m, \cdots)$. In this case, the enumerator $E(n_j, m, \cdots)$ is computed by setting parameter n to n_j and recursively calling the procedure to compute the formula for $E(m, \cdots) = E(n, m, \cdots) \mid_{n=n_0}$. This recursion will eventually terminate in case 1 when all parameters have been fixed. The recurrent procedure returns a pseudo-polynomial formula and thus the resulting $\vec{E}$ is a vector of pseudo-polynomials. The matrix-vector product $A^{-1}\vec{E}$ must therefore be done symbolically and procedures to perform symbolic multiplication and addition of pseudo-polynomials are used to compute $u_{j,k}$.

This entire procedure is then repeated for other sets of values for n, that have different values of $k = (n \bmod q)$, until all of the u_{jk} have been solved.

4.3 Simplifying pseudo-polynomials

The pseudo-polynomial returned by the above procedure is not usually in simplest form. A procedure called *Reduce* is called to reduce the pseudo-polynomial E. It is defined as follows:

Reduce(E)

1. If E is not a simple rational number, (it is a polynomial of degree $p-1$, or a periodic number of period p), then for $0 \leq j < p$ recursively call Reduce($E.u_j$).
2. If E is a periodic number ($E = [u_{p-1}, \cdots, u_0]$), then

(a) Try to reduce the period of E.
The period of E can be reduced from p to q if

$$\exists\, q \;:\; 1 \leq q < p;\; \frac{p}{q} \in \mathcal{Z} \left(\forall\, j, c \;:\; 0 \leq j < q;\; 0 \leq c < \frac{p}{q} \;\; (u_k = u_{cq+k}) \right)$$

(b) Try to reduce the strength of E.
If the period of E is one ($E = [u_0]$), reduce it from a periodic number to a simple (non-periodic) rational number u_0.

3. If E is a polynomial ($E = u_{p-1}x^{p-1} + u_{p-2}x^{p-2} + \cdots + u_0 x^0$), then

(a) Try to reduce the degree of E.
The degree of E can be reduced from $p-1$ to $q-1$ if

$$\exists\, q \;:\; 1 \leq q < p\, (\forall\, j \;:\; q \leq j < p \;\; (u_j = 0))$$

(b) Try to reduce the strength of E.
If the degree of E is zero ($E = u_0 x^0$), reduce it from a polynomial to a simple rational number u_0.

Example 7

$$\left[\frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{3}{4}\right]_n n^2 + \left[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right]_n n + \left[0, -\frac{1}{4}, 0, -\frac{1}{4}, 0, -\frac{1}{4}\right]_n$$

is reduced to

$$\frac{3}{4}n^2 + \frac{1}{2}n + \left[0, -\frac{1}{4}\right]_n$$

□

5 Examples and related work

All the examples considered in this section come from several related works [13, 14, 27, 26, 11, 21] and were also handled by W. Pugh in [21], except example 13 whose aim is to show the generality of our method.

Example 8

M. Haghighat and C. Polychronopoulos present a method for volume computation in [13, 14]. Their first example is $\sum_{i=1}^{n} \sum_{j=3}^{i} \sum_{k=j}^{5} 1$. This sum defines a set of linear constraints $P_N = \{1 \leq i \leq n, 3 \leq j \leq i, j \leq k \leq 5\}$. The Loechner/Wilde algorithm computes the following vertices for the polytope P_n:

- (1) if $3 \leq n \leq 5$, the vertices are $(3,3,5)$, $(3,3,3)$, $(n,3,5)$, $(n,3,3)$, $(n,n,5)$, (n,n,n).
- (2) if $n > 5$, the vertices are $(3,3,5)$, $(3,3,3)$, $(n,3,5)$, $(n,3,3)$, $(5,5,5)$, $(n,5,5)$.

Hence, P_n is an integral 3-polytope and the associated pseudo-polynomial $E_n = c_1 n^3 + c_2 n^2 + c_3 n + c_4$ is computed for each of the two domains.

Domain 1: From $n = 3$ to 5 we count $E_n = 3, 8, 14$. Since we have only three equations and four unknowns, we can arbitrarily set c_1 to zero, reducing the degree of the polynomial enumerator to 2. The remaining coefficients are solved for using the following system of equations:

$$(1) \begin{cases} c_1 & = 0 \\ 9c_2 + 3c_3 + c_4 & = E_3 = 3 \\ 16c_2 + 4c_3 + c_4 & = E_4 = 8 \\ 25c_2 + 5c_3 + c_4 & = E_5 = 14 \end{cases}$$

The solution of this system results in $E_n = \frac{1}{2}n^2 + \frac{3}{2}n - 6$.

Our result is different from that computed by Pugh [21], because Pugh only used two points $n = 3$ to 4 where as our algorithm used $n = 3$ to 5. Both are correct over their respective ranges of n.

Domain 2: From $n = 5$ to 8 we count $E_n = 14, 20, 26, 32$.

$$(2) \begin{cases} 125c_1 + 25c_2 + 5c_3 + c_4 & = E_5 = 16 \\ 216c_1 + 36c_2 + 6c_3 + c_4 & = E_6 = 20 \\ 343c_1 + 49c_2 + 7c_3 + c_4 & = E_7 = 26 \\ 512c_1 + 64c_2 + 8c_3 + c_4 & = E_8 = 32 \end{cases}$$

The solution of this system results in $E_n = 6n - 16$.

Summary The enumerator is thus

$$E_n = \begin{cases} \frac{1}{2}n^2 + \frac{3}{2}n - 6 & \text{if } 3 \le n \le 5 \\ 6n - 16 & \text{if } 5 \le n \end{cases}$$

This is equivalent to the result derived by W. Pugh. On the other hand, M. Haghighat and C. Polychronopoulos derived the enumerator:

$$\mu(\min(n-2,3))(-(\min(n,5))^3 + 15(\min(n,5))^2 - 38\min(n,5) + 24)/6 + 6\max(n-5,0)$$

where $\mu(x)$ is defined to be 1 if x is positive and 0 otherwise. The enumerator evaluates to the same values as ours, but the form is quite different, and more complicated, because of the $\mu(x)$, *min* and *max* functions they employ. Our method never generates $\mu(x)$, *min* or *max* functions.

□

Example 9

The second example in [13, 14] is

$$\sum_{i=1}^{2n} \sum_{j=1}^{min(i,2n-i)} 1$$

This sum defines a set of linear constraints $P_n = \{1 \le i \le 2n, 1 \le j \le i, i + j \le 2n\}$. The Loechner/Wilde algorithm computes the following vertices for the polytope P_n:

- if $n \ge 1$, the vertices are $(n, n), (2n - 1, 1), (1, 1)$.

Hence, P_n is an integral 2-polytope and the associated pseudo-polynomial enumerator $E_n = c_1 n^2 + c_2 n + c_3$ is formulated.

From $n = 1$ to 3 we count $E_n = 1, 4, 9$. We solve for the unknown coefficients using the following system of equations:

$$\begin{cases} c_1 + c_2 + c_3 & = E_1 = 1 \\ 4c_1 + 2c_2 + c_3 & = E_2 = 4 \\ 9c_1 + 3c_2 + c_3 & = E_3 = 9 \end{cases}$$

The solution of this system results in $E_n = n^2$, which is the same result as W. Pugh. In [13, 14], Haghighat and Polychronopoulos do not describe their technique in detail. They define a number of rules in order to transform expressions, but nothing is said on how to decide which rule to apply and when. As in [27], they assume the summation must be performed in a predetermined order.
□

Example 10

We now handle Pugh's "more elaborate" example of [21], and show that our method does not encounter any difficulty whatsoever, while Pugh had to introduce additional techniques which were not elaborated in his paper. This example is the set of linear constraints $P_n = \{1 \leq i;\ j \leq n;\ 2i \leq 3j\}$.

The Loechner/Wilde algorithm computes the following vertices for the so-defined polytope P_n:

- if $n \geq \frac{2}{3}$, the vertices are $(\frac{3n}{2}, n), (1, \frac{2}{3}), (1, n)$

Hence, P_n is a rational 2-polytope with a denominator of 6. The associated enumerator is a pseudo-polynomial of degree 2 and period 6:

$$E_n = [c_1, c_2, c_3, c_4, c_5, c_6]n^2 + [c_7, c_8, c_9, c_{10}, c_{11}, c_{12}]n + [c_{13}, c_{14}, c_{15}, c_{16}, c_{17}, c_{18}]$$

From $n = 1$ to 18 we count $E_n = 1$, 4, 8, 14, 21, 30, 40, 52, 65, 80, 96, 114, 133, 154, 176, 200, 225, 252. The 18 equations are solved as 6 systems of 3 equations a piece:

$$\begin{cases} c_2 + c_8 + c_{14} & = E_1 = 1 \\ 49c_2 + 7c_8 + c_{14} & = E_7 = 40 \\ 169c_2 + 13c_8 + c_{14} & = E_{13} = 133 \end{cases} \quad \begin{cases} 4c_3 + 2c_9 + c_{15} & = E_2 = 4 \\ 64c_3 + 8c_9 + c_{15} & = E_8 = 52 \\ 196c_3 + 14c_9 + c_{15} & = E_{14} = 154 \end{cases}$$

$$\begin{cases} 9c_4 + 3c_{10} + c_{16} & = E_3 = 8 \\ 81c_4 + 9c_{10} + c_{16} & = E_9 = 65 \\ 225c_4 + 15c_{10} + c_{16} & = E_{15} = 176 \end{cases} \quad \begin{cases} 16c_5 + 4c_{11} + c_{17} & = E_4 = 14 \\ 100c_5 + 10c_{11} + c_{17} & = E_{10} = 80 \\ 256c_5 + 16c_{11} + c_{17} & = E_{16} = 200 \end{cases}$$

$$\begin{cases} 25c_6 + 5c_{12} + c_{18} & = E_5 = 21 \\ 121c_6 + 11c_{12} + c_{18} & = E_{11} = 96 \\ 289c_6 + 17c_{12} + c_{18} & = E_{17} = 225 \end{cases} \quad \begin{cases} 36c_1 + 6c_7 + c_{13} & = E_6 = 30 \\ 144c_1 + 12c_7 + c_{13} & = E_{12} = 114 \\ 324c_1 + 18c_7 + c_{13} & = E_{18} = 252 \end{cases}$$

The solution of these systems results in

$$E_n = \left[\frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{3}{4}\right]_n n^2 + \left[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right]_n n + \left[-\frac{1}{4}, 0, -\frac{1}{4}, 0, -\frac{1}{4}, 0\right]_n$$

which can be directly simplified to

$$E_n = \frac{3}{4}n^2 + \frac{1}{2}n + \left[-\frac{1}{4}, 0\right]_n$$

Pugh [21] uses a different technique to derive an enumerator of

$$\begin{cases} \frac{(n+n \bmod 2)(3n+3(n \bmod 2)-2)}{8} & if\ 1 \leq n \\ \frac{3(n-n \bmod 2)(n-n \bmod 2+2)}{8} & if\ 2 \leq n \end{cases}$$

This expression is then intuitively simplified to $\frac{3n^2+2n-n \bmod 2}{4}$ which evaluates to the same values as our pseudo-polynomial enumerator. This example shows that Pugh's technique tended to complicate answers when considering rational polytopes, and hence, does not allow to consider systematically any set of rational linear constraints. Our method has no difficulty handling any set of rational linear constraints without any restriction.

The method described by Pugh in [21] consists of a set of tools and techniques dedicated to simplifying the initial set of constraints. Applying all of these tools could be expensive for large sets of constraints with rational coefficients. Moreover, their application is often rule driven, in some cases involving the addition of auxiliary variables and the use of standard formulas for sums of powers. (Pugh expects it will be sufficient to hard code the formulas for powers up to 10.) And finally, the resulting formula involves the use of modulo functions. This shows some of the limitations in the general use of Pugh's method.

□

6 Application to the analysis of nested-loop programs

In this section, we give several examples of how the method is applied to applications which need to analyze nested-loops.

Example 11

An example in [11] asks to calculate the number of memory locations touched in a Successive Over-Relaxation (SOR) code:

```
for i := 2 to n-1 do
  for j := 2 to n-1 do
      a(i,j) = (2*a(i,j) + a(i-1,j) + a(i+1,j)
                + a(i,j-1) + a(i,j+1)) / 6
```

The elements of `a` touched by this loop are

$$\begin{array}{rl} & \{(i,j)|2 \le i,j \le n-1\} \\ \cup & \{(i-1,j)|2 \le i,j \le n-1\} \\ \cup & \{(i+1,j)|2 \le i,j \le n-1\} \\ \cup & \{(i,j-1)|2 \le i,j \le n-1\} \\ \cup & \{(i,j+1)|2 \le i,j \le n-1\} \end{array} = \begin{array}{rl} & \{(i,j)|2 \le i,j \le n-1\} \\ \cup & \{(i,j)|1 \le i \le n-2, 2 \le j \le n-1\} \\ \cup & \{(i,j)|3 \le i \le n, 2 \le j \le n-1\} \\ \cup & \{(i,j)|2 \le i \le n-1, 1 \le j \le n-2\} \\ \cup & \{(i,j)|2 \le i \le n-1, 3 \le j \le n\} \end{array}$$

The parametric vertices of this union of polytopes P_n are computed by using the heuristic procedure described in subsection 4.1. It gives the following results: $(1,2)$, $(2,1)$, $(1,n-1)$, $(2,n)$, $(n-1,n)$, $(n,n-1)$, $(n-1,1)$ and $(n,2)$. Hence, it is an integral 2-polytope and the associated Ehrhart polynomial is $E_n = c_1 n^2 + c_2 n + c_3$.

From $n = 4$ to 6 we count $E_n = 12, 21, 32$.

$$\begin{cases} 16c_1 + 4c_2 + c_3 &= E_4 = 12 \\ 25c_1 + 5c_2 + c_3 &= E_5 = 21 \\ 36c_1 + 6c_2 + c_3 &= E_6 = 32 \end{cases}$$

The solution of this system gives the result $E_n = n^2 - 4$ which is the same answer that Pugh gave.

□

Example 12

An extension of our method to several size parameters has been implemented. It is applied to an example handled by Tawbi in [26] which is $\sum_{i=1}^{n}\sum_{j=1}^{i}\sum_{k=j}^{m} 1$. This sum defines a set of linear constraints $\{1 \leq i \leq n, 1 \leq j \leq i, j \leq k \leq m\}$. The Loechner/Wilde algorithm computes the following vertices for the so-defined polytope $P_{n,m}$:

- (1) if $1 \leq n \leq m$, the vertices are $(1,1,m)$, $(1,1,1)$, $(n,1,m)$, $(n,1,1)$, (n,n,m), (n,n,n).
- (2) if $1 \leq m \leq n$, the vertices are $(1,1,m)$, $(1,1,1)$, $(n,1,m)$, $(n,1,1)$, (m,m,m), (n,m,m).

Hence, $P_{n,m}$ is an integral 3-polytope and the associated pseudo-polynomial is of the form $E_{n,m} = \sum_{i=0}^{3}\sum_{j=0}^{3-i} c_{ij} n^i m^j$ and has to be computed for each of the two domains, (1) and (2).

Domain 1: Our method, applied to several size parameters, uses a symbolic resolution by first considering the polynomial $E_{n,m} = d_3(m)n^3 + d_2(m)n^2 + d_1(m)n + d_0(m)$ where:

$$\begin{array}{rcl} d_3(m) & = & c_{33}m^3 + c_{32}m^2 + c_{31}m + c_{30} \\ d_2(m) & = & c_{23}m^3 + c_{22}m^2 + c_{21}m + c_{20} \\ d_1(m) & = & c_{13}m^3 + c_{12}m^2 + c_{11}m + c_{10} \\ d_0(m) & = & c_{03}m^3 + c_{02}m^2 + c_{01}m + c_{00} \end{array}$$

Symbolically evaluating $E_{n,m}$ at $n = 1, 2, 3, 4$, yields the following symbolic system of equations.

$$\left\{ \begin{array}{ll} d_3(m) + d_2(m) + d_1(m) + d_0(m) & = E_{1,m} \\ 8d_3(m) + 4d_2(m) + 2d_1(m) + d_0(m) & = E_{2,m} \\ 27d_3(m) + 9d_2(m) + 3d_1(m) + d_0(m) & = E_{2,m} \\ 64d_3(m) + 16d_2(m) + 4d_1(m) + d_0(m) & = E_{4,m} \end{array} \right.$$

In order to solve for the symbolic coefficients $d_3(m)$, $d_2(m)$, $d_1(m)$, and $d_0(m)$, the enumerators $E_{1,m}$, $E_{2,m}$, $E_{3,m}$ and $E_{4,m}$ have to be computed.

For $n = 1$ and from $m = 1$ to 4, we count $E_{n,m} = 1, 2, 3, 4$.

$$\left\{ \begin{array}{ll} c_1 + c_2 + c_3 + c_4 & = E_{1,1} = 1 \\ 8c_1 + 4c_2 + 2c_3 + c_4 & = E_{1,2} = 2 \\ 27c_1 + 9c_2 + 3c_3 + c_4 & = E_{1,3} = 3 \\ 64c_1 + 16c_2 + 4c_3 + c_4 & = E_{1,4} = 4 \end{array} \right.$$

This system results in $E_{1,m} = m$. In the same way, we compute $E_{2,m} = 3m - 1$, $E_{3,m} = 6m - 4$ and $E_{4,m} = 10m - 10$, giving the equations

$$\left\{ \begin{array}{ll} d_3(m) + d_2(m) + d_1(m) + d_0(m) & = m \\ 8d_3(m) + 4d_2(m) + 2d_1(m) + d_0(m) & = 3m - 1 \\ 27d_3(m) + 9d_2(m) + 3d_1(m) + d_0(m) & = 6m - 4 \\ 64d_3(m) + 16d_2(m) + 4d_1(m) + d_0(m) & = 10m - 10 \end{array} \right.$$

The expressions for $d_1(m), d_2(m), d_3(m)$ and $d_4(m)$ can be symbolically computed as: $d_1 = -\frac{1}{6}$, $d_2 = \frac{m}{2}$, $d_3 = \frac{m}{2} + \frac{1}{6}$ and $d_4 = 0$. Thus, for domain $1 \leq n \leq m$, the enumerator $E_{n,m}$ is derived as:

$$\begin{array}{rcl} E_{n,m} & = & -\frac{1}{6}n^3 + \frac{m}{2}n^2 + (\frac{1}{2}m + \frac{1}{6})n + 0 \\ & = & -\frac{n^3}{6} + \frac{n^2 m}{2} + \frac{nm}{2} + \frac{n}{6} \end{array}$$

Domain 2: The same work is done for domain (2), i.e. $1 \leq m \leq n$, resulting in $E_{n,m} = -\frac{m^3}{6} + \frac{m^2 n}{2} + \frac{mn}{2} + \frac{m}{6}$, which matches the result of Pugh. Our method can be applied the same way when considering several size parameters in the case of rational polytopes, case which is not systematically handled by Pugh's techniques. Tawbi in [27, 26] first used a polyhedral splitting technique to characterize the definition domains, and an elimination of the variables in a predetermined order. However, her method does not allow systematically computing exact formulae like our method.
□

Example 13

We propose a more complex example than any previously published, requiring the use of the entire method we have described. This example involves rational parameterized polytopes depending on three size parameters. Let us consider the following loop nest:

```
for i := 0 to n do
  for j := 0 to i + m/2 do
    for k := 0 to i - n + p do
      a(i,j,k) = a(i,j,k-1) + a(k,j,k)*a(i,k,k)
```

We want to compute the number of flops in order to evaluate the execution time of this code segment. This loop nest is modeled by the convex polytope $P_N = \{z \in \mathcal{Z}^3 \mid Az \leq BN + C\}$ where

$$A = \begin{pmatrix} -1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \\ -2 & 2 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{pmatrix} \quad N = \begin{pmatrix} n \\ m \\ p \end{pmatrix} \quad C = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

The Loechner/Wilde algorithm computes the following vertices for the so-defined polytope P_N:
- (1) if $n \geq p$: $(n-p, \frac{2n+m-2p}{2}, 0)$, $(n-p, 0, 0)$, $(n, \frac{2n+m}{2}, p)$, $(n, \frac{2n+m}{2}, 0)$, $(n, 0, p)$, $(n, 0, 0)$.
- (2) if $n \leq p$: $(0, \frac{m}{2}, -n+p)$, $(0, \frac{m}{2}, 0)$, $(0, 0, -n+p)$, $(0, 0, 0)$, $(n, \frac{2n+m}{2}, p)$, $(n, \frac{2n+m}{2}, 0)$, $(n, 0, p)$, $(n, 0, 0)$.

As an example, we now compute the pseudo-polynomial enumerator on the first definition domain.

The denominator of P_N is 2. We count $((k+1)d)^q = 512$ initial values of $E(n, m, p)$. The lowest initial value of n is 9 since we consider the first definition domain ($n \geq p$) and since for any initial value of n, 8 initial values of p have to be considered. For example, with $n = 9$ we get the following initial counts:

n=9 p / m	1	2	3	4	5	6	7	8
1	29	56	90	130	175	224	276	330
2	32	62	100	145	196	252	312	375
3	32	62	100	145	196	252	312	375
4	35	68	110	160	217	280	348	420
5	35	68	110	160	217	280	348	420
6	38	74	120	175	238	308	384	465
7	38	74	120	175	238	308	384	465
8	41	80	130	190	259	336	420	510

Each line of this array allows resolving a system of equations to determine $E(x, y, p)$, for all $x \in [9..16]$ and $y \in [1..8]$. For example, the first system dedicated to the determination of $E(9, 1, p) = [c_2, c_1]_p\, n_3^3 + [c_4, c_3]_p\, n_3^2 + [c_6, c_5]_p\, n_3 + [c_8, c_7]_p$ is

$$n = 9, m = 1 : \begin{cases} p = 1 : & c_1 + c_3 + c_5 + c_7 & = 29 \\ p = 2 : & 8c_2 + 4c_4 + 2c_6 + c_8 & = 56 \\ p = 3 : & 27c_1 + 9c_3 + 3c_5 + c_7 & = 90 \\ p = 4 : & 64c_2 + 16c_4 + 4c_6 + c_8 & = 130 \\ p = 5 : & 125c_1 + 25c_3 + 5c_5 + c_7 & = 175 \\ p = 6 : & 216c_2 + 36c_4 + 6c_6 + c_8 & = 224 \\ p = 7 : & 343c_1 + 49c_3 + 7c_5 + c_7 & = 276 \\ p = 8 : & 512c_2 + 64c_4 + 8c_6 + c_8 & = 330 \end{cases}$$

resulting in $E(9, 1, p) = -\frac{1}{6}p^3 + \frac{9}{2}p^2 + \frac{44}{3}p + 10$.

Solving for the other polynomials, $E(9, 2, p) \ldots E(9, 8, p)$, allows resolving systems to determine $E(9, m, p) = [c_2, c_1]_m\, m^3 + [c_4, c_3]_m\, m^2 + [c_6, c_5]_m\, m + [c_8, c_7]_m$ with this system of equations:

$$n = 9 : \begin{cases} m = 1 : & c_1 + c_3 + c_5 + c_7 & = -\frac{1}{6}p^3 + \frac{9}{2}p^2 + \frac{44}{3}n_3 + 10 \\ m = 2 : & 8c_2 + 4c_4 + 2c_6 + c_8 & = -\frac{1}{6}p^3 + 5p^2 + \frac{97}{6}p + 11 \\ m = 3 : & 27c_1 + 9c_3 + 3c_5 + c_7 & = -\frac{1}{6}p^3 + 5p^2 + \frac{97}{6}p + 11 \\ m = 4 : & 64c_2 + 16c_4 + 4c_6 + c_8 & = -\frac{1}{6}p^3 + \frac{11}{2}p^2 + \frac{53}{3}p + 12 \\ m = 5 : & 125c_1 + 25c_3 + 5c_5 + c_7 & = -\frac{1}{6}p^3 + \frac{11}{2}p^2 + \frac{53}{3}p + 12 \\ m = 6 : & 216c_2 + 36c_4 + 6c_6 + c_8 & = -\frac{1}{6}p^3 + 6p^2 + \frac{115}{6}p + 13 \\ m = 7 : & 343c_1 + 49c_3 + 7c_5 + c_7 & = -\frac{1}{6}p^3 + 6p^2 + \frac{115}{6}p + 13 \\ m = 8 : & 512c_2 + 64c_4 + 8c_6 + c_8 & = -\frac{1}{6}p^3 + \frac{13}{2}p^2 + \frac{62}{3}p + 14 \end{cases}$$

In this case, the following result is obtained:

$$E(9, m, p) = \tfrac{1}{4}mp^2 + \tfrac{3}{4}mp + \tfrac{1}{2}m - \tfrac{1}{6}p^3 + \left[\tfrac{9}{2}, \tfrac{17}{4}\right]_m p^2 + \left[\tfrac{44}{3}, \tfrac{167}{12}\right]_m p + \left[10, \tfrac{19}{2}\right]_m$$

In a similar way, $E(10, m, p)$, $E(11, m, p)$, $\ldots$, $E(16, m, p)$ are computed. Finally, $E(n, m, p)$ is determined by resolving the system shown below:

$$\begin{cases} n = 9 : & 729c_1 + 81c_3 + 9c_5 + c_7 & = & \frac{1}{4}mp^2 + \frac{3}{4}mp + \frac{1}{2}m - \frac{1}{6}p^3 + [\frac{9}{2}, \frac{17}{4}]_m p^2 + [\frac{44}{3}, \frac{167}{12}]_m p + [10, \frac{19}{2}]_m \\ n = 10 : & 1000c_2 + 100c_4 + 10c_6 + c_8 & = & \frac{1}{4}mp^2 + \frac{3}{4}mp + \frac{1}{2}m - \frac{1}{6}p^3 + [5, \frac{19}{4}]_m p^2 + [\frac{97}{6}, \frac{185}{12}]_m p + [11, \frac{21}{2}]_m \\ n = 11 : & 1331c_1 + 121c_3 + 11c_5 + c_7 & = & \frac{1}{4}mp^2 + \frac{3}{4}mp + \frac{1}{2}m - \frac{1}{6}p^3 + [\frac{11}{2}, \frac{21}{4}]_m p^2 + [\frac{53}{3}, \frac{203}{12}]_m p + [12, \frac{23}{2}]_m \\ n = 12 : & 1728c_2 + 144c_4 + 12c_6 + c_8 & = & \frac{1}{4}mp^2 + \frac{3}{4}mp + \frac{1}{2}m - \frac{1}{6}p^3 + [6, \frac{23}{4}]_m p^2 + [\frac{115}{6}, \frac{221}{12}]_m p + [13, \frac{25}{2}]_m \\ n = 13 : & 2197c_1 + 169c_3 + 13c_5 + c_7 & = & \frac{1}{4}mp^2 + \frac{3}{4}mp + \frac{1}{2}m - \frac{1}{6}p^3 + [\frac{13}{2}, \frac{25}{4}]_m p^2 + [\frac{62}{3}, \frac{239}{12}]_m p + [14, \frac{27}{2}]_m \\ n = 14 : & 2744c_2 + 196c_4 + 14c_6 + c_8 & = & \frac{1}{4}mp^2 + \frac{3}{4}mp + \frac{1}{2}m - \frac{1}{6}p^3 + [7, \frac{27}{4}]_m p^2 + [\frac{133}{6}, \frac{257}{12}]_m p + [15, \frac{29}{2}]_m \\ n = 15 : & 3375c_1 + 225c_3 + 15c_5 + c_7 & = & \frac{1}{4}mp^2 + \frac{3}{4}mp + \frac{1}{2}m - \frac{1}{6}p^3 + [\frac{15}{2}, \frac{29}{4},]_m p^2 + [\frac{71}{3}, \frac{275}{12}]_m p + [16, \frac{31}{2}]_m \\ n = 16 : & 4096c_2 + 256c_4 + 16c_6 + c_8 & = & \frac{1}{4}mp^2 + \frac{3}{4}mp + \frac{1}{2}m - \frac{1}{6}p^3 + [8, \frac{31}{4}]_m p^2 + [\frac{151}{6}, \frac{293}{12}]_m p + [17, \frac{33}{2}]_m \end{cases}$$

The following result is obtained, giving the number of flops executed by the loop nest when $n \geq p$:

$$E(n, m, p) = \frac{1}{2}np^2 + \frac{3}{2}np + n + \frac{1}{4}mp^2 + \frac{3}{4}mp + \frac{1}{2}m - \frac{1}{6}p^3 + \left[0, -\frac{1}{4}\right]_m p^2 + \left[\frac{7}{6}, \frac{5}{12}\right]_m p + \left[1, \frac{1}{2}\right]_m$$

Likewise, when $n \leq p$:

$$E(n, m, p) = -\tfrac{1}{6}n^3 - \tfrac{1}{4}n^2m + n^2\left[\tfrac{1}{2}p, \tfrac{1}{2}p - \tfrac{1}{4}\right]_m + \tfrac{1}{2}nmp + \tfrac{1}{4}nm + n\left[\tfrac{3}{2}p + \tfrac{7}{6}, p + \tfrac{11}{12}\right]_m + \tfrac{1}{2}mp + \tfrac{1}{2}m + \left[p + 1, \tfrac{1}{2}p + \tfrac{1}{2}\right]_m$$

□

Example 14

For our last example, we show an application of parametric analysis. We derive the maximum amount of parallelism found in a small nested-loop program.

Consider the following loop nest:

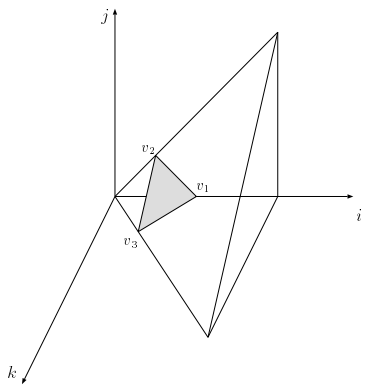


Figure 3: T_{n,t_0} in the first validity domain

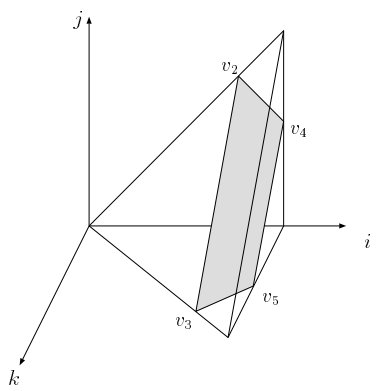


Figure 4: T_{n,t_0} in the second validity domain

```
for i:= 0 to n do
  for j:= 0 to i do
    for k:= 0 to i-j do
      Statement(s)
```

This loop nest is modeled by the one-parameter polyhedral iteration space defined by:

$$P_n = \{\begin{pmatrix} i \\ j \\ k \end{pmatrix} \in \mathcal{Z}^3 | \begin{pmatrix} -1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 1 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ k \end{pmatrix} \leq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} n\}$$

Assume that the dependence analysis of this loop nest yields an optimal linear schedule defined by $t(i,j,k) = i + j + k$: any iteration (i,j,k) is computed at instant $i + j + k$. We use our method to determine the parametric number of simultaneous calculations $par(n, t_0)$ occuring at any instant t_0, in order to determine the parametric maximum parallelism $maxpar(n) = MAX_{t_0}\{par(n, t_0)\}$, i.e. the maximum number of simultaneous calculations.

From a geometrical point of view, all the points (i,j,k) of P_n resulting in the same value of $t_0 = t(i,j,k)$, belong to the two-parameters rational 2-polytope T_{n,t_0} defined by $T_{n,t_0} = P_n \cap \{(i,j,k) \in \mathcal{Z}^3 | i + j + k = t_0\}$. The number of points in T_{n,t_0} is $par(n, t_0)$.

Giving this last parametric definition of T_{n,t_0} as an input to the parametric vertices finding algorithm, it outputs the following results as shown on figures 3 and 4:

- if $0 \leq t_0 \leq n$, the parametric vertices are $v_1 = (t_0, 0, 0)$, $v_2 = \left(\frac{t_0}{2}, \frac{t_0}{2}, 0\right)$ and $v_3 = \left(\frac{t_0}{2}, 0, \frac{t_0}{2}\right)$.
- if $n \leq t_0 \leq 2n$, the parametric vertices are $v_2 = \left(\frac{t_0}{2}, \frac{t_0}{2}, 0\right)$, $v_3 = \left(\frac{t_0}{2}, 0, \frac{t_0}{2}\right)$, $v_4 = (n, t_0 - n, 0)$ and $v_5 = (n, 0, t - n)$.
- otherwise, T_{n,t_0} does not exist and $par(n, t_0) = 0$.

Hence, $par(n, t_0)$ is defined on two validity domains as pseudo-polynomials in n and t_0, of degree 2, and of pseudo-period 2. According to theorem 3 and by looking at $T_{1,0}$, it is deduced that all the coefficients of n^q, $q \leq 2$, are constant: $par(n, t_0) = c_1 n^2 + c_2 n + c_3$.

(i) We determine c_1, c_2, c_3 by solving the following system of equations on the first validity domain:

$$\begin{cases} 36c_1 + 6c_2 + c_3 &= par(6, t_0) \\ 49c_1 + 7c_2 + c_3 &= par(7, t_0) \\ 64c_1 + 8c_2 + c_3 &= par(8, t_0) \end{cases}$$

Solving this system yields the equations for $par(6, t_0)$, $par(7, t_0)$ and $par(8, t_0)$. According to theorem 3 and by looking at $T_{1,1}$, it is deduced that only the coefficient of t_0^2 is constant. For example, $par(6, t_0) = c_1 t_0^2 + [c_3, c_2]t_0 + [c_5, c_4]$, such that:

$$\begin{cases} c_5 &= par(6, 0) = 1 \\ c_1 + c_2 + c_4 &= par(6, 1) = 1 \\ 4c_1 + 2c_3 + c_5 &= par(6, 2) = 3 \\ 9c_1 + 3c_2 + c_4 &= par(6, 3) = 3 \\ 16c_1 + 4c_3 + c_5 &= par(6, 4) = 6 \end{cases}$$

where the values of $par(6, 0), \ldots, par(6, 4)$ are given by some initial counts.
$\Rightarrow par(6, t_0) = \frac{1}{8}t_0^2 + [\frac{3}{4}, \frac{1}{2}]t_0 + [1, \frac{3}{8}]$. In the same way, we determine that the other pseudo-polynomials are the same, i.e $\forall\ n \in [6..8],\ par(n, t_0) = \frac{1}{8}t_0^2 + [\frac{3}{4}, \frac{1}{2}]t_0 + [1, \frac{3}{8}]$. Solving the first system gives the parallelism rate of any instant t_0 on the first validity domain:

$$\forall\ 0 \leq t_0 \leq n,\ par(n, t_0) = \tfrac{1}{8}t_0^2 + \left[\tfrac{3}{4}, \tfrac{1}{2}\right] t_0 + \left[1, \tfrac{3}{8}\right]$$

Observe that the parallelism rate does not depend on n in this validity domain.
(ii) We now consider the second validity domain.

$$\begin{cases} 36c_1 + 6c_2 + c_3 &= par(6, t_0) \\ 49c_1 + 7c_2 + c_3 &= par(7, t_0) \\ 64c_1 + 8c_2 + c_3 &= par(8, t_0) \end{cases}$$

Solving this system yields the equations for $par(6, t_0)$, $par(7, t_0)$ and $par(8, t_0)$. For example, $par(6, t_0) = c_1 t_0^2 + [c_3, c_2]t_0 + [c_5, c_4]$, such that:

$$\begin{cases} 36c_1 + 6c_3 + c_5 &= par(6, 6) &= 10 \\ 49c_1 + 7c_2 + c_4 &= par(6, 7) &= 9 \\ 64c_1 + 8c_3 + c_5 &= par(6, 8) &= 12 \\ 81c_1 + 9c_2 + c_4 &= par(6, 9) &= 9 \\ 100c_1 + 10c_3 + c_5 &= par(6, 10) &= 11 \end{cases}$$

$\Rightarrow par(6, t_0) = -\frac{3}{8}t_0^2 + [\frac{25}{4}, 6]t_0 + [-14, -\frac{117}{8}]$. In the same way, we determine the other pseudo-polynomials:

$$\begin{aligned} par(7, t_0) &= -\tfrac{3}{8}t_0^2 + [\tfrac{29}{4}, 7]t_0 + [-20, -\tfrac{165}{8}] \\ par(8, t_0) &= -\tfrac{3}{8}t_0^2 + [\tfrac{33}{4}, 8]t_0 + [-27, -\tfrac{221}{8}] \end{aligned}$$

Solving the first system gives the parallelism rate of any instant t_0 on the second validity domain:

$$\forall\ n \leq t_0 \leq 2n, par(n, t_0) = nt_0 - \tfrac{1}{2}n^2 + \tfrac{1}{2}n - \tfrac{3}{8}t_0^2 + \left[\tfrac{1}{4}, 0\right]_{t_0} t_0 - \left[1, \tfrac{3}{8}\right]_{t_0}$$

(iii) From these results, the maximum parallelism is deduced:

- On the first validity domain ($0 \leq t_0 \leq n$), $par(n, t_0)$ is maximum when t_0 is maximum, i.e. when $t_0 = n$. Hence, $maxpar(n) = \frac{1}{8}n^2 + \left[\frac{3}{4}, \frac{1}{2}\right] n + \left[1, \frac{3}{8}\right]$.

- On the second validity domain ($n \leq t_0 \leq 2n$), we take the derivative of $par(n, t_0)$ with respect to t_0: $par'(n, t_0) = n - \frac{3}{4}t_0 + \left[\frac{1}{4}, 0\right]_{t_0}$.

$$\begin{array}{rl} & par'(n, t_0) \leq 0 \\ \Leftrightarrow & n - \frac{3}{4}t_0 + \left[0, \frac{1}{4}\right]_{t_0} \leq 0 \\ \Leftrightarrow & t_0 \geq \frac{4n-[1,0]}{3} \end{array}$$

  Hence, $par(n, t_0)$ increases from $t_0 = n$ to $t_0 = \frac{4n-[1,0]}{3}$, and then decreases until $t_0 = 2n$: it is maximum for $t_0 = t_{max} = \frac{4n-[1,0]}{3}$. Since t_{max} is not always integral, a case study is necessary to get the exact value of $maxpar(n) = par(n, t_{max})$, checking the possible values of $\lfloor t_{max} \rfloor$ and $\lceil t_{max} \rceil$. It yields the following answer: $maxpar(n) = \frac{1}{6}n^2 + \frac{5}{6}n - \left[1, 1, \frac{23}{24}\right]$.

- In conclusion, the maximum parallelism of the loop nest is:

$$\forall\ n \geq 0,\ maxpar(n) = \frac{1}{6}n^2 + \frac{5}{6}n - \left[1, 1, \frac{23}{24}\right]$$

□

7 Non-linear extensions

All these results concern normal polyhedra defined by a combination of linear constraints. When considering some nonlinear ones, the transformation rules given in this section have to be applied first.

This set of constraints depends on some positive integral size parameters. Any constraint C_k is of the form

$$C_k : \sum_i a_i \alpha_i \leq \sum_j b_j n_j + c$$

where the a_i's, b_i's and c are rational constants, the n_j's are the size parameters and the α_i's can be of the form

$$x_i \quad (x_i\ mod\ d_i) \quad \lfloor \frac{x_i}{d_i} \rfloor \quad \lceil \frac{x_i}{d_i} \rceil \quad (d_i | x_i)$$

where d_i is an integral constant, $(d_i|x_i)$ means that d_i evenly divides x_i and x_i is any rational linear combination of the free variables. When the form of any α_i yields a nonlinear constraint, this can easily be transformed into a linear one through some elementary transformations, as those presented by W. Pugh in [21]:

• If a term $\lfloor \frac{x_i}{d_i} \rfloor$ appears in C_k, a free variable z is added to the set of free variables, and constraint C_k is replaced with $\{d_i z \leq x_i < d_i(z+1), C'_k\}$ where C'_k is C_k with $\lfloor \frac{x_i}{d_i} \rfloor$ replaced with z.

• If a term $\lceil \frac{x_i}{d_i} \rceil$ appears in C_k, a free variable z is added to the set of free variables, and constraint C_k is replaced with $\{d_i(z-1) < x_i \leq d_i z, C'_k\}$ where C'_k is C_k with $\lceil \frac{x_i}{d_i} \rceil$ replaced with z.

• If a term $(x_i\ mod\ d_i)$ appears in C_k, a free variable z is added to the set of free variables, and constraint C_k is replaced with $\{d_i z \leq x_i < d_i(z+1), C'_k\}$ where C'_k is C_k with $(x_i\ mod\ d_i)$ replaced with $x_i - z$.

• A constraint $(d_i|x_i)$ is equivalent to $(x_i \ mod \ d_i = 0)$ and the precedent rule can be applied.
• If a term of the form $\lfloor \frac{U}{u} \rfloor$ appears in the right side of the constraint C_k, since rational constraints are considered by our method, the floor can be avoided and the whole constraint is multiplied by u.

8 Conclusion

We have presented a general method, based on the vertices of parameterized polytopes and Ehrhart polynomials, for deriving a symbolic formula for the number of integer solutions to a set of parameterized linear constraints. The enumerator is formulated symbolically as a pseudo-polynomial in terms of parameters, The method we have presented is robust and handles any set of rational parameterized linear constraints without any restriction. Moreover, the method gives a closed form formula which easily evaluates to exact counts. The procedure is entirely computable and has been implemented along with the parametric vertices finding algorithm of Loechner and Wilde [17] and operates in association with the *polyhedral library* [28].

Being able to find symbolic enumerators has many applications in the analysis and transformation of scientific programs. We have given several examples in this paper. There are many other applications in the analysis and transformation of loop nests, some of which can be found in works cited here above or in [4, 5]. Other applications may also be found in other fields as sequential program analysis, combinatorics and optimization.

We have compared our method with other similar methods. M. Haghighat and C. Polychronopoulos present in [13, 14] a method for volume computation. The form of their answers is different because of the *min* and *max* expressions they introduce. Their results tend to be more complicated.

Ferrante, Sarkar and Thrash in [11] present methods for computing the number of distinct memory locations and cache lines accessed by a loop nest. These methods cannot handle unions of polytopes, do not compute symbolic answers, often compute only an approximation, and use expensive methods to handle the cache lines touched by a set of references.

The method described by Pugh in [21] consists of a set of tools and techniques dedicated to simplifying the initial set of constraints. Applying all of these tools could be expensive for large sets of constraints with rational coefficients. Moreover, their application is often rule driven, in some cases involving the addition of auxiliary variables and the use of standard formulas for sums of powers. (Pugh expects it will be sufficient to hard code the formulas for powers up to 10.) And finally, the resulting formula involves the use of modulo functions.

Tawbi in [27, 26] first uses a polyhedral splitting technique to characterize the validity domains, and an elimination of the variables in a predetermined order. However, her method does not systematically compute exact answers.

After doing these comparisons, we make these observations:

- summations over several variables should not presume an order in which to perform the summation (as pointed out by Pugh in [21]),
- a general technique must also consider rational polytopes, since nested loops models often yield rational affine loop bounds, and since it is useful to systematically handle a wide variety of applications in the parallelization of scientific programs.

- in order to be simple, the result should be expressed without any complex function as *min*'s, *max*'s, *mod*'s, floors or ceilings. In this respect, periodic numbers, introduced by Ehrhart [10], are a good alternative and can undoubtedly be used in other situations.
- exact formulations can be found.

This is the first method to be proposed that enumerates both integer and rational polytopes using a single comprehensive algorithm. Moreover, the formulation, using Ehrhart pseudo-polynomials, is simple and gives exact parametric expressions for the number of integer points.

References

[1] A. I. Barvinok. A polynomial-time algorithm for counting integral points in polyhedra when the dimension is fixed. In *Proc. of the 34th Symp. on the Foundations of Computer Science (FOCS'93)*, pages 566-572. IEEE Computer Society Press, New York, 1993.

[2] A. I. Barvinok. Computing the Ehrhart Polynomial of a Convex Lattice Polytope. *Discrete Comput. Geom.*, 12:35-48,1994.

[3] M.S. Bazaraa, H.D. Sherali and C.M. Shetty. *Nonlinear Programming, Theory and Algorithms, second edition*. Wiley, New York, 1993.

[4] Ph. Clauss. The volume of a lattice polyhedron to enumerate processors and parallelism. Research Report ICPS 95-11, 1995. `http://icps.u-strasbg.fr/pub-95/pub-95-11.ps.gz`

[5] Ph. Clauss. Counting Solutions to Linear and Nonlinear Constraints through Ehrhart polynomials: Applications to Analyze and Transform Scientific Programs. *10th ACM Int. Conf. on Supercomputing*, Philadelphia, May 1996. Also available as Research Report ICPS 95-20. `http://icps.u-strasbg.fr/pub-96/pub-95-20.ps.gz`

[6] M. Dyer and A. M. Frieze. On the complexity of computing the volume of a polyhedron. *SIAM J. Comput.*, 17(5):967-974, 1988.

[7] E. Ehrhart. Sur les polyèdres rationnels homothétiques à n dimensions. *C.R. Acad. Sci. Paris*, 254:616-618, 1962.

[8] E. Ehrhart. Sur un problème de géométrie diophantienne linéaire I. *J. Reine Angew. Math.*, 226:1-29, 1967.

[9] E. Ehrhart. Sur un problème de géométrie diophantienne linéaire II. *J. Reine Angew. Math.*, 227:25-49, 1967.

[10] E. Ehrhart. *Polynômes arithmétiques et Méthode des Polyèdres en Combinatoire*. International Series of Numerical Mathematics, vol.35, Birkhäuser Verlag, Basel/Stuttgart, 1977.

[11] J. Ferrante, V. Sarkar and W. Thrash. On estimating and enhancing cache effectiveness. *Advances in Languages and Compilers for Parallel Processing*, pages 328-343. The MIT Press, 1991.

[12] P.M. Gruber and C.G. Lekkerkerker. *Geometry of Numbers*. North-Holland, Amsterdam, 1987.

[13] M. Haghighat and C. Polychronopoulos. Symbolic analysis: A basis for parallelization, optimization and scheduling of programs. *U. Banerjee et al., editor, Languages and Compilers for Parallel Computing*. Springer-Verlag, August 1993. LNCS 768, *Proc. of the 6th annual workshop on Programming Languages and Compilers for Parallel Computing*.

[14] M. Haghighat and C. Polychronopoulos. Symbolic analysis: A basis for parallelization, optimization and scheduling of programs. Technical Report 1317, CSRD, Univ. of Illinois, August 1993.

[15] D. J. Kuck. *The Structure of Computers and Computations*, J. Wiley and Sons, N.Y., 1978.

[16] H. Le Verge, V. Van Dongen, and D. Wilde. La synthèse de nids de boucles avec la bibliothèque polyédrique. *RenPar'6*, Jun 1994. English version "Loop Nest Synthesis Using the Polyhedral Library" in IRISA technical report 830, May 1994.

[17] V. Loechner and D. K. Wilde. Parameterized polyhedra and their vertices. Research Report ICPS 96-09, Submitted for publication, 1996. `http://icps.u-strasbg.fr/pub-96/pub-96-09.ps.gz`

[18] I. G. Macdonald. The volume of a lattice polyhedron. *Proc. Camb. Phil. Soc.*, 59:719-726, 1963.

[19] I. G. Macdonald. Polynomials associated with finite cell-complexes. *J. London Math. Soc.*, 4(2):181-192, 1971.

[20] T. H. Mattheiss and D. Rubin. A survey and comparison of methods for finding all vertices of convex polyhedral sets. *Mathematics of Operations Research*, 5(2):167-185, 1980.

[21] W. Pugh. Counting Solutions to Presburger Formulas: How and Why. *Proc. of the 1994 ACM SIGPLAN Conference on Programming Language Design and Implementation*, 1994.

[22] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, New York, 1986.

[23] H. S. Stone and D. Thiebaut. Footprints in the cache. *Proc. ACM SIGMETRICS 1986*, pp. 4-8, May 1986.

[24] R. P. Stanley. *Combinatorics and Commutative Algebra*. Birkhäuser, Boston, 1983.

[25] R. P. Stanley. *Enumerative Combinatorics Volume I*. Wadsworth & Brooks/Cole, Monterey, California, 1986.

[26] N. Tawbi. Estimation of nested loop execution time by integer arithmetics in convex polyhedra. *Proc. of the 1994 International Parallel Processing Symposium*, April 1994.

[27] N. Tawbi and P. Feautrier. Processor allocation and loop scheduling on multiprocessor computers. *Proc. of the 1992 International Conference on Supercomputing*, pages 63-71, July 1992.

[28] D. K. Wilde. A library for doing polyhedral operations. *Master's thesis*, Oregon State University, Corvallis, Oregon, Dec 1993. Also published as IRISA technical report PI 785, Rennes, France, Dec 1993.