On the calculation of Ehrhart polynomials in degenerate domains

Bavo Nootaert, Kristof Beyls, Erik D'Hollander

January 27, 2005

Abstract

Many program analyses require the computation of the number of integral solutions to a parametric system of linear constraints. This count is represented by a number of Ehrhart quasi-polynomials, each one valid on a specific domain in the parameter space.

One of the algorithms to calculate Ehrhart quasi-polynomials cannot be used when the validity domains have a particular geometric form. In this note, we show that a suitable homothetic transformation of the problem can avoid these degenerate domains, thus making the algorithm generally applicable.

1 Introduction

Many program analyses and compiler optimizations are based on a mathematical representation of the program called the *polyhedral model*. A typical program that benefits from this model consists of nested loops and data organized in arrays. Both the loop bounds and the array indices are affine expressions of the induction variables. In the polyhedral model, a set of linear inequalities is constructed that represents some characteristic of the program, such as the iteration space corresponding to a loop or the cache lines an array reference accesses during the execution of the program. Frequently the number of points that satisfy the inequalities has to be counted. This count may depend on several parameters. Currently, there exist two methods for computing this function. Clauss's method [CL98] has some limitations that prevent it from being applicable in all cases. Our extension to this method will allow it to always compute the desired function.

```
for (i=0; i<10; i++)
    for (j=0; j<N; j++)
        f(a[j],i);
```

Figure 1: Example loop

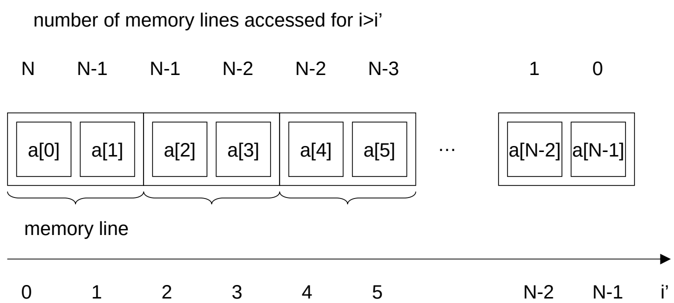


Figure 2: Number of memory lines accessed after iteration i'

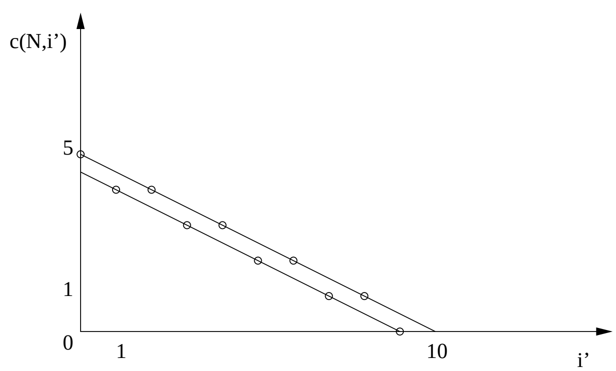


Figure 3: $c(N, i')$ for $N = 10$, marked by $\circ$. The alternative representation by linear functions which are evaluated periodically is also shown.

1.1 The polyhedral model: an example

Consider the loop in Figure 1. Denote by $@a[\mathtt{j}]$ the memory address of $a[\mathtt{j}]$. For every value of j, $@a[\mathtt{j}]$ is accessed 10 times: once for each i-iteration. Suppose the program is executed on a machine equipped with a fully associative LRU cache, and each cache line can hold exactly two array elements. The *memory line* of $a[\mathtt{j}]$ is the block of memory that contains $@a[\mathtt{j}]$ and maps precisely to a single cache line. For simplicity, assume $@a[0] = 0$. Then the memory line of $a[\mathtt{j}]$ is $\left\lfloor \frac{@a[\mathtt{j}]}{2} \right\rfloor$. In some analyses to model cache use, a formula is required that gives the number of memory lines accessed between one access to a memory line, and the next access. This is called the *forward reuse distance*. If a memory line is never accessed again, the forward reuse distance is infinite. In this example, the forward reuse distance of $@a[\mathtt{j}]$ is $\lceil \frac{N}{2} \rceil$, for each access in the iterations $\{(\mathtt{i},\mathtt{j})|0 \leq \mathtt{i} < 9\}$. Let us now try to find this formula using the polyhedral model. Fix an iteration (I, J). In this iteration, only $@a[J]$ is accessed. The next use of $@a[J]$ is in iteration $(I+1, J)$. Clearly the forward reuse distance is independent of I. We need to count the memory lines accessed by all iterations between a use and the next reuse. This is the set

$$\left\{ \left\lfloor \frac{@a[j]}{2} \right\rfloor \mid (i = I \text{ and } J < j < N) \text{ or } (i = I+1 \text{ and } 0 \leq j < J) \right\}.$$

Because the polyhedral model only handles conjunctions of linear inequalities, and no disjunctions, this set is split up into the disjoint sets $\left\{ \left\lfloor \frac{@a[j]}{2} \right\rfloor \mid i = I \text{ and } J < j < N \right\}$ and $\left\{ \left\lfloor \frac{@a[j]}{2} \right\rfloor | i = I+1 \text{ and } 0 \leq j < J \right\}$, which are further simplified into respectively $\{k|J \leq 2k < N\}$ and $\{k|0 \leq 2k < J\}$. Denote the number of elements in these sets by $c(N, J)$ and $d(N, J)$, respectively. Then the reuse distance is $c(N, J) + d(N, J)$.

For example $c(N, J)$ is depicted in Figure 3, for $N = 10$. $c(N, J)$ can be written as $c(N, J) = \lceil \frac{N}{2} \rceil - \lceil \frac{J}{2} \rceil$. An alternative way of writing this is by a set of linear functions:

$$\begin{array}{ll} c_0^0(N,J) = \frac{N}{2} - \frac{J}{2} & c_1^0(N,J) = \frac{N}{2} + \frac{1}{2} - \frac{J}{2} \\ c_0^1(N,J) = \frac{N}{2} - \frac{J}{2} - \frac{1}{2} & c_1^1(N,J) = \frac{N}{2} - \frac{J}{2} \end{array}$$

Then

$$c(N,J) = \begin{cases} c_0^0(N,J) & \text{if } N \bmod 2 = 0 \text{ and } J \bmod 2 = 0 \\ c_0^1(N,J) & \text{if } N \bmod 2 = 0 \text{ and } J \bmod 2 = 1 \\ c_1^0(N,J) & \text{if } N \bmod 2 = 1 \text{ and } J \bmod 2 = 0 \\ c_1^1(N,J) & \text{if } N \bmod 2 = 1 \text{ and } J \bmod 2 = 1 \end{cases}$$

In general, a function like c that cycles through a set of polynomials depending on its arguments, is called a *quasi-polynomial*. A rigorous definition is given in Section 2.

2 Mathematical representation

2.1 Polyhedra

We first recall some basic definitions from integer polyhedral theory [Sch86, CL98, LW97]. In this paper, vectors are denoted by bold symbols. $\mathbf{2}^X$ denotes the power set of a set X. A **(convex) polyhedron** P is the intersection of a finite number of halfspaces. Symbolically this can written as $P = \{\mathbf{x} \in \mathbb{Q}^d | A\mathbf{x} \geq \mathbf{b}\}$, where $\mathbf{b} \in \mathbb{Z}^l$ and $A \in \mathbb{Z}^{l\times d}$. P can be bounded or unbounded. A bounded convex polyhedron is called a **(convex) polytope**. We write $\# P$ for the number of integer points in a polytope P, i.e. the number of elements of the set $\{P \cap \mathbb{Z}^d\}$.[1] A **supporting hyperplane** of P is a hyperplane A such that P lies entirely in a closed half space determined by A, and $A \cap P \neq \emptyset$. The **affine span** of P is the smallest (w.r.t. $\subseteq$) affine subspace W of $\mathbb{Q}^d$ such that $P \subseteq W$. The **dimension** of P is the dimension of its affine span (notation: $\dim P$). A **face** of P is $\emptyset$, P, or the intersection of P and a supporting hyperplane. Faces are polyhedra too. Faces of dimension 0, 1 and $\dim P - 1$ are called **vertices**, **edges** and **facets**, respectively.

To be able to represent problems that depend on a number of parameters, these concepts are extended as follows [CL98, LW97]. A **parametric (convex) polyhedron** is a function $Q : \mathbb{Q}^k \to \mathbf{2}^{\mathbb{Q}^d}$ defined by $Q(\mathbf{n}) = \{\mathbf{x} \in \mathbb{Q}^d | A\mathbf{x} \geq B\mathbf{n} + \mathbf{b}\}$, for each $\mathbf{n} \in \mathbb{Q}^k$. If $Q(\mathbf{n})$ is bounded for each $\mathbf{n} \in \mathbb{Q}^k$, it is called a **parametric (convex) polytope**. Intuitively, a parametric polytope is a polytope where some facets are allowed to move in the direction along their normal. Note that generally, facets cannot move independently: a translation of one facet (i.e. a different value for the parameters) can cause a translation of another facet.

To any function $T : \mathbb{Q}^k \to \mathbf{2}^{\mathbb{Q}^d}$ corresponds a set $\widehat{T}$ in the **combined data and parameter space** defined by $\widehat{T} = \{(\mathbf{x}, \mathbf{n})^T | \mathbf{x} \in T(\mathbf{n})\}$. If Q is a parametric polyhedron, then $\widehat{Q}$ is a non-parametric polyhedron. The projections π^i and π_i take the first i and the last i elements of a vector, respectively. In this paper, we assume without loss of generality that $\dim \pi_k \widehat{Q} = k$. If this is not the case, then there exist $\boldsymbol{\alpha}$ and β such that $\boldsymbol{\alpha} \cdot \mathbf{n} = \beta$, and we can eliminate parameters via affine substitutions. A **parametric vertex** of Q is a function v, such that $v(\mathbf{n}) = \pi^d(F \cap \{(\mathbf{x}, \mathbf{n})^T | \mathbf{x} \in \mathbb{Q}^d\})$, for a k–dimensional face F of $\widehat{Q}$, and $v(\mathbf{n})$ is nonempty for each $\mathbf{n} \in \operatorname{dom} v$. Loechner and Wilde [LW97] showed that $v(\mathbf{n})$ is a vertex[2] of $Q(\mathbf{n})$, the domain of v is $\pi_k F$, and v is an affine function.

Example 2.1 Consider the 2–dimensional parametric polytope Q, defined by $Q(m, n) = \{(i, j)^T \mid 0 \leq i \leq m, 0 \leq j \leq m, 2i + 2j \leq n\}$. $\widehat{Q}$ is the 4–dimensional polytope $\{(i, j, m, n)^T | 0 \leq i \leq m, 0 \leq j \leq m, 2i + 2j \leq n\}$. The parametric vertices are computed with the Loechner-Wilde algorithm, implemented in PolyLib [Loe99]:

$$\begin{array}{ll}
(0,0) & \text{defined for all } (0,0) \leq (m,n) \\
(m,0),(0,m) & \text{defined if } 0 \leq 2m \leq n \\
(\frac{n}{2},0),(0,\frac{n}{2}) & \text{defined if } 0 \leq n \leq 2m \\
(-m+\frac{n}{2},m),(m,-m+\frac{n}{2}) & \text{defined if } 0 \leq 2m \leq n \leq 4m \\
(m,m) & \text{defined if } 0 \leq 4m \leq n
\end{array}.$$

[1] We could have defined polyhedra as a set of *integer* points, subject to linear constraints. However, by working in $\mathbb{Q}^d$, we will be able to use all the convenient properties of vector spaces.

[2] More accurately, $v(\mathbf{n})$ is a set with a single point, which is a vertex.

2.2 Ehrhart quasi-polynomials

A function $U : \mathbb{Z}^k \to X$ is **periodic** if there exists a $\mathbf{p} = (p_1, \ldots, p_n)^T \in \mathbb{N}^k$ such that $U(\mathbf{n}) = U(\mathbf{n}')$ whenever $n_i \equiv n' \mod p_i$. If $k = 1$, U can be represented by a p-tuple $a_0, \ldots, a_{p-1}$, with $U(n) = a_{n \bmod p}$. We write $U(n) = [a_0, \ldots, a_{p-1}]_n$. A function $f : \mathbb{Z}^k \to \mathbb{Q}$ is a **quasi-polynomial** of degree d and pseudoperiod $\mathbf{p} \in \mathbb{N}^k$ if it can be written as

$$f(\mathbf{n}) = \sum_{i_1+i_2+\ldots i_k \leq d} U_{i_1,i_2,\ldots,i_k}(n_1, n_2, \ldots, n_k) n_1^{i_1} n_2^{i_2} \ldots n_k^{i_k},$$

where $U_{i_1,i_2,\ldots,i_k}$ are $\mathbf{p}$-periodic functions.

Example 2.2 Define

$$U(n) = \begin{cases} 2 & \text{if } n \bmod 4 = 0 \\ 2 & \text{if } n \bmod 4 = 1 \\ 5 & \text{if } n \bmod 4 = 2 \\ 3 & \text{if } n \bmod 4 = 3 \end{cases}$$

Then U is a periodic number of period 4 and $U(n) = [2, 2, 5, 3]_n$. The function $f(n) = [2, 2, 5, 3]_n n^2 + [1, 2, 3, 6]_n n + [1, 7, 5, 6]_n$ is a pseudo-polynomial.

Clauss showed that there exists a subdivision of the parameter space, such that over each region, the number of points in a parametric polytope is expressed by a pseudo-polynomial. These regions are called *validity domains*, and are closely related to the parametric vertices.

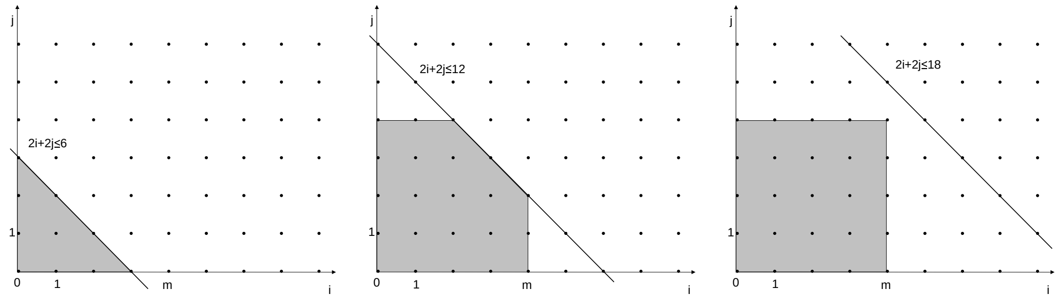
Definition 2.1 *Let Q be a parametric polytope. Consider the set X of subsets of $\operatorname{dom} Q$, defined by: $V \in X$ if for each parametric vertex v of Q, $\operatorname{dom} v \cap V = \emptyset$ or $\operatorname{dom} v \cap V = V$. An element of X that is maximal w.r.t. $\subseteq$ is called a* ***validity domain****.*

Clauss and Loechner have given an algorithm to compute validity domains in [CL98]. Moreover, they showed that the dimension of a validity domain is equal to the number of parameters.

Next, we need the following concepts. The **denominator** of a $q \in \mathbb{Q}$ is the smallest $l \in \mathbb{N}$ such that $lq \in \mathbb{Z}$. The **denominator** of a parametric vertex v is the vector $\mathbf{p} = (p_1, \ldots, p_k)^T$, with p_i the denominator of the coefficient of n_i in the affine expression that defines v. If the coefficient of n_i is zero, we define $p_i = 1$. The **denominator** of a parametric polyhedron in a validity domain V is the vector $\mathbf{p} = (p_1, \ldots, p_k)^T$, with p_i the least common multiple of the i–th components of the denominators of the parametric vertices defined in V. The following theorem was discovered by Ehrhart [Ehr67] for one parameter, extended by Clauss to more than one parameter, and refined by Verdoolaege et al. [VSB$^+$04], giving a sharper upper bound on the degree.

Theorem 2.1 (Ehrhart-Clauss) *There exists a unique mapping $\mathcal{E}$, such that for each parametric polytope Q, validity domain V of Q and $\mathbf{n} \in V \cap \mathbb{Z}^k$:*

- *$\mathcal{E}(Q, V)$ is a quasi-polynomial with at most the dimension of Q as degree and the denominator of Q as pseudo-period,*



(a) (b) (c)

Figure 4: The parametric polytope from Example 2.3, for $m = 4$, and (a) $n = 6$, (b) $n = 12$, (c) $n = 18$.

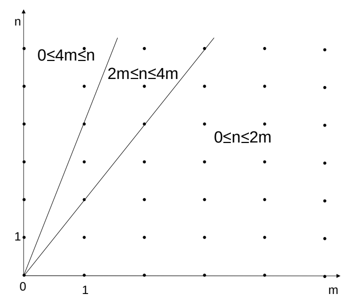


Figure 5: The validity domains from Example 2.3

- $\mathcal{E}(Q, V)(\mathbf{n}) = \# Q(\mathbf{n})$.

The first clause in this theorem leads to the following definition:

Definition 2.2 (Ehrhart quasi-polynomial) *Let Q be a parametric polytope and V a validity domain of Q, and $\mathcal{E}$ the mapping given by theorem 2.1. $\mathcal{E}(Q, V)$ is called the* ***Ehrhart quasi-polynomial*** *of Q in V.*

Example 2.3 Consider again the parametric polytope from Example 2.1. We rewrite the list of parametric vertices of Q as follows:

$$\begin{cases} (\frac{n}{2}, 0), (0, \frac{n}{2}), (0, 0) & \text{if } 0 \leq n \leq 2m \\ (-m + \frac{n}{2}, m), (m, -m + \frac{n}{2}), (m, 0), (0, m), (0, 0) & \text{if } 0 \leq 2m \leq n \leq 4m \\ (m, m), (m, 0), (0, m), (0, 0) & \text{if } 0 \leq 4m \leq n \end{cases} .$$

Thus Q has three validity domains (Figure 5), each with a corresponding set of parametric vertices. For each validity domain an instance of the parametric polytope is depicted in Figure 4.

The Ehrhart quasi-polynomials are

$$\begin{cases} \frac{1}{8}n^2 + [\frac{3}{4}, \frac{1}{2}]_n n + [1, \frac{3}{8}]_n & \text{if } 0 \leq n \leq 2m \\ -m^2 + nm + -\frac{1}{8}n^2 + [1, 0]_n m + [\frac{1}{4}, \frac{1}{2}]_n n + [1, \frac{5}{8}]_n & \text{if } 2m \leq n \leq 4m \\ m^2 + 2m + 1 & \text{if } 0 \leq 4m \leq n \end{cases} .$$

3 Clauss's method

Clauss and Loechner [CL98] developed an algorithm to compute the Ehrhart quasi-polynomials, based on the information provided by Theorem 2.1. In this section we explain the algorithm, and indicate where it fails.

3.1 One parameter

Let Q be a parametric polytope of dimension d with 1 parameter and denominator p, and V a validity domain of Q. It follows from Theorem 2.1 that $\mathcal{E}(Q,V)(n)$ can be written as

$$[c_{0,d}, c_{1,d}, \ldots, c_{p-1,d}]_n n^d + [c_{0,d-1}, c_{1,d-1}, \ldots, c_{p-1,d-1}]_n n^{d-1} + \ldots + [c_{0,0}, c_{1,0}, \ldots, c_{p-1,0}]_n, \quad (1)$$

where the $c_{i,j}$, with $0 \le i < p$ and $0 \le j \le d$, are yet unknown. To determine each $c_{i,j}$ it suffices to calculate $\# Q(n)$ for $(d+1)p$ consecutive values of n (say $\nu_0, \ldots, \nu_{(d+1)p-1}$) in $V \cap \mathbb{Z}^k$, and solve the following systems, obtained by substituting $\nu_0, \ldots, \nu_{(d+1)p-1}$ in (1), for each i, $0 \le i < p$:

$$\begin{pmatrix} \nu_i^d & \nu_i^{d-1} & \cdots & 1 \\ \nu_{i+p}^d & \nu_{i+p}^d & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ \nu_{i+dp}^d & \nu_{i+dp}^d & \cdots & 1 \end{pmatrix} \begin{pmatrix} c_{i,d} \\ c_{i,d-1} \\ \vdots \\ c_{i,0} \end{pmatrix} = \begin{pmatrix} \# Q(\nu_i) \\ \# Q(\nu_{i+p}) \\ \vdots \\ \# Q(\nu_{i+dp}) \end{pmatrix}. \quad (2)$$

Since the matrix formed by the powers of the parameter values is a Vandermonde matrix, this system always has a unique solution, *provided that sufficient values can be found in the validity domain.*[3] Note that if there are at least $(d+1)p$ integer points in V, consecutive values can be chosen since V is convex

3.2 More than one parameter

The idea of Clauss's method is to compute the Ehrhart quasi-polynomial of a parametric polytope with more than one parameter *recursively*. Let Q be a parametric polytope with $k > 1$ parameters $\mathbf{n} = (n_1, \ldots, n_{k-1}, n_k)$, and pseudo-period $\mathbf{p} = (p_1, \ldots, p_{k-1}, p_k)$. Analogously to the case with one parameter, $(d+1)p_k$ consecutive values of n_k are chosen, denoted by $\nu_0, \ldots, \nu_{(d+1)p_k-1}$. For these values $\# Q(n_1, \ldots, n_{k-1}, \nu_j)$ is computed. However, since $n_1, \ldots, n_{k-1}$ are parameters, $\# Q(n_1, \ldots, n_{k-1}, \nu_j)$ are quasi-polynomials themselves, which are determined recursively. Next, a set of systems analogous to (2) is solved, where the right-hand side is a vector of quasi-polynomials. Symbolically, we write $\mathcal{E}(Q,V)(n_1, \ldots, n_k)$ as

$$[e_{0,d}, e_{1,d}, \ldots, e_{p-1,d}]_{n_k} n_k^d + [e_{0,d-1}, e_{1,d-1}, \ldots, e_{p-1,d-1}]_{n_k} n_k^{d-1} + \ldots + [e_{0,0}, e_{1,0}, \ldots, e_{p-1,0}]_{n_k}, \quad (3)$$

[3] Recall that formula (1) is only valid for parameter values in the validity domain, which may be bounded.

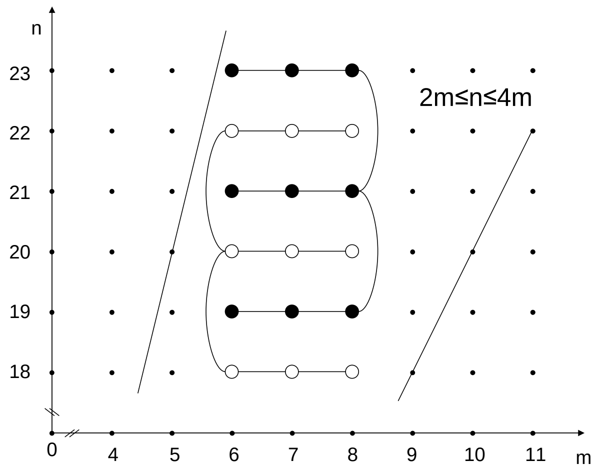


Figure 6: Parameter values in the validity domain $\{(m,n)|2m \leq n \leq 4m\}$ used to compute the Ehrhart quasi-polynomial in Example 3.1. The points marked by $\circ$ determine the coefficients for $n \equiv 0 \mod 2$, and those marked by $\bullet$ determine the coefficients for $n \equiv 1 \mod 2$.

where the $e_{i,j}$ are quasi-polynomials, with parameters $n_1, \ldots, n_{k-1}$ (which are omitted here for typographical reasons). The system (2) becomes:

$$\begin{pmatrix} \nu_i^d & \nu_i^{d-1} & \cdots & 1 \\ \nu_{i+p}^d & \nu_{i+p}^d & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ \nu_{i+dp}^d & \nu_{i+dp}^d & \cdots & 1 \end{pmatrix} \begin{pmatrix} e_{i,d} \\ e_{i,d-1} \\ \vdots \\ e_{i,0} \end{pmatrix} = \begin{pmatrix} \# Q(n_1, \ldots, n_{k-1}, \nu_i) \\ \# Q(n_1, \ldots, n_{k-1}, \nu_{i+p}) \\ \vdots \\ \# Q(n_1, \ldots, n_{k-1}, \nu_{i+dp}) \end{pmatrix}. \tag{4}$$

Remark that since the matrix of coefficients contains only numbers, a standard algorithm suffices to solve this system. Addition of two quasi-polynomials, and scalar multiplication of a quasi-polynomial, are the standard extensions of these operations to functions. As with regular polynomials, this means just adding the terms of equal degree, resp. multiplying all coefficients with the scalar. Addition and scalar multiplication of periodic functions represented as vectors are done componentwise.

Example 3.1 Recall the parametric polytope from Example 2.3. Using Clauss's method, we compute the Ehrhart quasi-polynomial in validity domain $V = \{(m,n)|2m \leq n \leq 4m\}$. By Theorem 2.1, the pseudo-period of $\mathcal{E}(P,V)$ is $(1,2)$. Hence, $\mathcal{E}(Q,V)(m,n) = [a,b]_n m^2 + ([c,d]_n n + [g,h]_n)m + [e,f]_n n^2 + [i,j]_n n + [k,l]_n$. For the recursive algorithm, this is rewritten as

$$\mathcal{E}(Q,V)(m,n) = e_1(n)m^2 + e_2(n)m + e_3(n), \tag{5}$$

where e_1, e_2 and e_3 are pseudo-polynomials with pseudo-period at most 2 and degree at most 2. We pick $m = 6, 7, 8$, which results in the following system:

$$\begin{cases} 36e_1(n) + 6e_2(n) + e_3(n) = \mathcal{E}(Q,V)(6,n) \\ 49e_1(n) + 7e_2(n) + e_3(n) = \mathcal{E}(Q,V)(7,n) \\ 64e_1(n) + 8e_2(n) + e_3(n) = \mathcal{E}(Q,V)(8,n) \end{cases}. \tag{6}$$

We now have to compute the right-hand side, for example $\mathcal{E}(Q,V)(6,n)$. This is a pseudo-polynomial of degree at most 2, with pseudo-period at most 2. It can be written as $\mathcal{E}(Q,V)(6,n) = [c_0,c_1]_n n^2 + [c_2,c_3]_n n + [c_4,c_5]_n$. Two systems are obtained, one for $n = 18,20,22$ to determine c_0, c_2 and c_4, and one for $n = 19,21,23$ to determine c_1, c_3 and c_5:

$$\begin{cases} 324c_0 + 18c_2 + c_4 = \mathcal{E}(Q,V)(6,18) \\ 400c_0 + 20c_2 + c_4 = \mathcal{E}(Q,V)(6,20) \\ 484c_0 + 22c_2 + c_4 = \mathcal{E}(Q,V)(6,22) \end{cases}, \quad \begin{cases} 361c_1 + 19c_3 + c_5 = \mathcal{E}(Q,V)(6,19) \\ 441c_1 + 21c_3 + c_5 = \mathcal{E}(Q,V)(6,21) \\ 529c_1 + 23c_3 + c_5 = \mathcal{E}(Q,V)(6,23) \end{cases}.$$

We get

$$\mathcal{E}(Q,V)(6,n) = \left[-\frac{1}{8},-\frac{1}{8}\right]_n n^2 + \left[\frac{25}{4},\frac{13}{2}\right]_n n + \left[-29,-\frac{283}{8}\right]_n,$$

and analogously,

$$\begin{aligned} \mathcal{E}(P,V)(7,n) &= \left[-\frac{1}{8},-\frac{1}{8}\right]_n n^2 + \left[\frac{29}{4},\frac{15}{2}\right]_n n + \left[-41,-\frac{387}{8}\right]_n \\ \mathcal{E}(Q,V)(8,n) &= \left[-\frac{1}{8},-\frac{1}{8}\right]_n n^2 + \left[\frac{33}{4},\frac{17}{2}\right]_n n + \left[-55,-\frac{507}{8}\right]_n \end{aligned}$$

We can now solve system (6) directly for $e_1(n)$, $e_2(n)$ and $e_3(n)$ using Gauss-elimination.

$$\begin{aligned} & \begin{cases} 36e_1(n) + 6e_2(n) + e_3(n) = \left[-\frac{1}{8},-\frac{1}{8}\right]_n n^2 + \left[\frac{25}{4},\frac{13}{2}\right]_n n + \left[-29,-\frac{283}{8}\right]_n \\ 49e_1(n) + 7e_2(n) + e_3(n) = \left[-\frac{1}{8},-\frac{1}{8}\right]_n n^2 + \left[\frac{29}{4},\frac{15}{2}\right]_n n + \left[-41,-\frac{387}{8}\right]_n \\ 64e_1(n) + 8e_2(n) + e_3(n) = \left[-\frac{1}{8},-\frac{1}{8}\right]_n n^2 + \left[\frac{33}{4},\frac{17}{2}\right]_n n + \left[-55,-\frac{507}{8}\right]_n \end{cases} \\ \equiv & \begin{cases} 36e_1(n) + 6e_2(n) + e_3(n) = \left[-\frac{1}{8},-\frac{1}{8}\right]_n n^2 + \left[\frac{25}{4},\frac{13}{2}\right]_n n + \left[-29,-\frac{283}{8}\right]_n \\ 13e_1(n) + e_2(n) = [1,1]_n n + [-12,-13]_n \\ 28e_1(n) + 2e_2(n) = [2,2]_n n + [-26,-28]_n \end{cases} \\ \equiv & \begin{cases} 36e_1(n) + 6e_2(n) + e_3(n) = \left[-\frac{1}{8},-\frac{1}{8}\right]_n n^2 + \left[\frac{25}{4},\frac{13}{2}\right]_n n + \left[-29,-\frac{283}{8}\right]_n \\ 13e_1(n) + e_2(n) = [1,1]_n n + [-12,-13]_n \\ 2e_1(n) = [-2,-2]_n \end{cases} \\ \equiv & \ldots \\ \equiv & \begin{cases} e_1(n) = [-1,-1]_n \\ e_2(n) = [1,1]_n n + [1,0]_n \\ e_3(n) = \left[-\frac{1}{8},-\frac{1}{8}\right]_n n^2 + \left[\frac{1}{4},\frac{1}{2}\right]_n n + \left[1,\frac{5}{8}\right]_n \end{cases}. \end{aligned}$$

Substituting these values in (5), we obtain

$$-m^2 + nm + [1,0]_n m + -\frac{1}{8}n^2 + \left[\frac{1}{4},\frac{1}{2}\right]_n n + \left[1,\tfrac{5}{8}\right]_n,$$

as in Example 2.3. The choice of parameter values is shown graphically in Figure 6.

Throughout the computation, only systems of the form (2) are encountered, which all have a unique solution. In each recursion however, $(d+1)p_i$ consecutive values have to be found. In Polylib, this is achieved by using points in a *hyperrectangle*.

```
for (i=M; i<=N+1; i++)
    for (j=N; j<=i+2; j++)
        S;
```

Figure 7: Example loop

Definition 3.1 (hyperrectangle, integer size vector) *A* ***hyperrectangle*** *in $\mathbb{Q}^d$ is a cartesian product of d bounded closed intervals in $\mathbb{Q}$. The* ***integer size vector*** $\mathbf{s}$ *of a hyperrectangle is the vector in $\mathbb{Z}^d$ for which the i–th element is the number of integer points in the interval in the i–th dimension.*

If a hyperrectangle with the required integer size vector does not exist, the validity domain is called 'degenerate', and the algorithm fails to compute the Ehrhart quasi-polynomial.

Definition 3.2 (degenerate domain) *A* ***degenerate domain*** *V is a validity domain of Q such that there exists no hyperrectangle $R \subseteq V$ with integer size vector $\mathbf{s} = ((d+1)p_1, (d+1)p_2, \ldots, (d+1)p_k)^T$.*

The implementation of Clauss's method in Polylib uses some heuristics to circumvent the problem of degenerate domains. These heuristics are not covering all situations: in several applications ([Bey04, CPHL01, TKD02, SVBL04]), degeneracy is an occurring problem. We have found many degenerate domains can be avoided by using Polylib's concept of a *context domain*. The context is a set of parameter values. Constraints from the parametric polytope that are redundant for all parameter values in that context are removed. In many cases this can allow interpolation using parameter values outside the validity domain. This is not generally true however.

Example 3.2 Consider the parametric polytope $Q(n) = \{(i,j)|0 \leq i, 0 \leq j \leq 2, 2i + 2j \leq n\}$, which has two validity domains: $V_1 = \{n|n \leq 4\}$ and $V_2 = \{n|n \geq 4\}$. In V_1, Q has vertices $(0,0)^T$, $(\frac{n}{2},0)^T$ and $(0,\frac{n}{2})^T$, so $\mathcal{E}(Q,V_1)$ has pseudo-period 2, and degree 2. Thus we need 6 values for n to compute the coefficients, while V_1 only contains 5 integer points. However, for $n \in V_1$, the constraint $j \leq 2$ is redundant. It follows that $\mathcal{E}(Q,V_1)$ is equal to $\mathcal{E}(Q',V)$, with $Q'(n) = \{(i,j)|0 \leq i, 0 \leq j, 2i + 2j \leq n\}$, and V the single validity domain of Q'.

Example 3.3 Consider the program in Figure 7. Suppose we want to know how many times statement S is executed in the loop. This will depend on the values of the parameters M and N. The possible values of the induction variables i and j are the integer points in the set $Q(M,N) = \{(i,j)^T|M \leq i \leq N+1 \text{ and } N \leq j \leq i+2\}$. Q has a degenerate domain $\{(M,N)| - 2 \leq M - N \leq 1\}$ (Figure 8), that is *not* resolved by supplying the validity domain as a context.

In the next section we describe our extension to Clauss's method, that can always compute the Ehrhart quasi-polynomial.

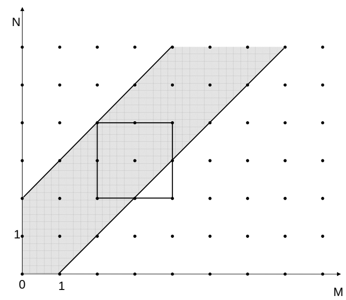


Figure 8: A degenerate domain: there exists no rectangle with integer size vector (3,3) that fits in the domain

4 Transformation to avoid degenerate domains

We first give the intuitive ideas behind our method, and then go on proving them mathematically.

The parameters of a parametric polytope allow a facet to 'move' in the direction along its normal. Validity domains limit the parameters to values where the parametric polytope has a particular shape. Changing the parameters to different values within the same domain, transforms the polytope without changing the structure of the polytope, i.e. the number of faces of each dimension and their relation to each other. TODO: ?i.e. Face lattice

Degenerate domains occur when a facet has not enough freedom to move: a number m parameter values are required, but the facet cannot move m steps without structural change of the polytope.

Our idea is to inflate the polytope, giving each facet enough freedom.

Consider the following transformation: $(\bar{H}Q)(\mathbf{n}) = \{\mathbf{x} \in \mathbb{Q} | A\mathbf{x} \geq B\mathbf{n} + h\mathbf{b}\}$, with $h \in \mathbb{Q}$. $\bar{H}$ is homothetic on the polyhedron in the combined data and parameter space.

Since validity domains are constructed form the polyhedron in the combined data and parameter space, $\bar{H}$ induces a homothetic transformation on the validity domains. Remark that if we set $h = 1$, we do not change the parametric polytope.

Thus using $\bar{H}$, we can transform the parametric polytope to one with larger validity domains. However, we are not directly interested in the Ehrhart quasi-polynomials of the transformed polytope, but in those of the original one. In fact, the Ehrhart quasi-polynomials of $\bar{H}Q$ count the number of rational points $\mathbf{x}$ in $Q(h^{-1}\mathbf{n})$, such that $h\mathbf{x}$ is integral. To be able to undo the transformation, we consider h a parameter, just like $\mathbf{n}$. Then the Ehrhart quasi-polynomials will also have an extra parameter, requiring more coefficients to be computed. Since the Ehrhart-polynomials are valid for all parameter values in a transformed domain, they are certainly valid for those points in the validity domain with $h = 1$. So the Ehrhart quasi-polynomials of Q are obtained from those of $\bar{H}Q$ by setting $h = 1$.

We now prove our method in four steps:

1. We define the transformation and give some of its basic properties.

2. We show that the induced transformation on the validity domains is homothetic as well.

3. We prove that the transformed validity domains are never degenerate.

4. We show how to compute the Ehrhart quasi-polynomials of the original parametric polytope from the Ehrhart quasi-polynomials of the transformed polytope.

4.1 Homothetic transformation

For a subset X of $\mathbb{Q}^d$ and an $h \in \mathbb{Q}$, denote by hX the set $\{h\mathbf{x} | \mathbf{x} \in X\}$. Define the mapping $H : \mathbf{2}^{\mathbb{Q}^d} \to \mathbf{2}^{\mathbb{Q}^{d+1}}$ such that for any subset X of $\mathbb{Q}^d$, $HX = \{(\mathbf{x}, h)^T \in \mathbb{Q}^d \times \mathbb{Q}_0^+ | \mathbf{x} \in hX\}$. Define $\bar{H} : (\mathbb{Q}^k \to \mathbb{Q}^d) \to (\mathbb{Q}^{k+1} \to \mathbb{Q}^d)$ by $(\bar{H}Q)(\mathbf{n}, h)^T = \{\mathbf{x} | (\mathbf{x}, \mathbf{n}, h)^T \in H\widehat{Q}\}$, again with $h > 0$. Then $\widehat{\bar{H}Q} = H\widehat{Q}$.

Essentially, $\bar{H}$ multiplies the constant term of Q by the parameter h:

$$
\begin{aligned}
\bar{H}Q(\mathbf{n}, h)^T &= \{\mathbf{x} \in \mathbb{Q}^d | (\mathbf{x}, \mathbf{n}, h)^T \in H\widehat{Q}\} \\
&= \{\mathbf{x} \in \mathbb{Q}^d | (\mathbf{x}, \mathbf{n})^T \in h\widehat{Q}\} \\
&= \{\mathbf{x} \in \mathbb{Q}^d | h^{-1}(\mathbf{x}, \mathbf{n})^T \in \widehat{Q}\} \\
&= \{\mathbf{x} \in \mathbb{Q}^d | Ah^{-1}\mathbf{x} \geq Bh^{-1}\mathbf{n} + \mathbf{b}\} \\
&= \{\mathbf{x} \in \mathbb{Q}^d | A\mathbf{x} \geq B\mathbf{n} + h\mathbf{b}\}
\end{aligned}
$$

Next, we prove some basic properties of H.

Lemma 4.1 *H distributes over $\cap$ and $\cup$: for any two subsets P_1 and P_2 of $\mathbb{Q}^d$, we have that*

$$H(P_1 \cap P_2) = HP_1 \cap HP_2$$

and

$$H(P_1 \cup P_2) = HP_1 \cup HP_2$$

Proof. We prove the lemma for $\cap$, the case $\cup$ being completely analogous.

$$
\begin{aligned}
(\mathbf{x}, h)^T \in H(P_1 \cap P_2) &\equiv h^{-1}\mathbf{x} \in P_1 \cap P_2 \\
&\equiv h^{-1}\mathbf{x} \in P_1 \wedge h^{-1}\mathbf{x} \in P_2 \\
&\equiv \mathbf{x} \in hP_1 \wedge \mathbf{x} \in hP_2 \\
&\equiv (\mathbf{x}, h)^T \in HP_1 \wedge (\mathbf{x}, h)^T \in HP_2 \\
&\equiv (\mathbf{x}, h)^T \in HP_1 \cap HP_2
\end{aligned}
$$

□

Lemma 4.2 *H commutes with projection onto the parameter space: for a parametric polytope Q, we have that*

$$\pi_{k+1}(H\widehat{Q}) = H(\pi_k\widehat{Q})$$

Proof.

$$
\begin{aligned}
(\mathbf{n},h)^T \in \pi_{k+1}(H\widehat{Q}) &\equiv \exists \mathbf{x}.(\mathbf{x},\mathbf{n},h)^T \in H\widehat{Q} && \{\text{by definition of } \pi_{k+1}\} \\
&\equiv \exists \mathbf{x}.h^{-1}(\mathbf{x},\mathbf{n})^T \in \widehat{Q} && \{\text{by definition of } H\} \\
&\equiv \exists \mathbf{y}.(\mathbf{y},h^{-1}\mathbf{n})^T \in \widehat{Q} && \{\text{change of dummy variable } \mathbf{y} = h^{-1}\mathbf{x}\} \\
&\equiv h^{-1}\mathbf{n} \in \pi_k\widehat{Q} && \{\text{by definition of } \pi_k\} \\
&\equiv (\mathbf{n},h)^T \in H(\pi_k\widehat{Q}) && \{\text{by definition of } H\}
\end{aligned}
$$

□

4.2 Induced transformation of the validity domains

As explained in Section 2, the validity domains of a parametric polytope with k parameters are formed by partitioning the projections of k-dimensional faces of the polyhedron in combined data and parameter space [LW97]. We first prove that a face remains a face under transformation, but the dimension is incremented by 1.

Lemma 4.3 *If F is a face of P, then HF is a face of HP.*

Proof. Let S be a supporting hyperplane of P defined by $\boldsymbol{\alpha}\cdot\mathbf{x} = \beta$, such that for all $\mathbf{x} \in P$, $\boldsymbol{\alpha}\cdot\mathbf{x} \le \beta$ and $S \cap P \ne \emptyset$. Then for all $(\mathbf{x},h)^T \in HP$, we have

$$
\begin{aligned}
(\mathbf{x},h)^T \in HP &\equiv h^{-1}\mathbf{x} \in P && \{\text{by definition of } H\} \\
&\Rightarrow \boldsymbol{\alpha}\cdot h^{-1}\mathbf{x} \le \beta && \{\text{since } S \text{ is a supporting hyperplane of } P\} \\
&\equiv \boldsymbol{\alpha}\cdot\mathbf{x} \le h\beta
\end{aligned}
$$

So HP lies entirely on one side of HS, which makes HS a supporting hyperplane of HP. By lemma 4.1, $HF = HS \cap HP$. We conclude that HF is a face of HP. □

Lemma 4.4 *For any polyhedron P,* $\dim HP = \dim P + 1$

For the proof, we need three more basic definitions. A point $\mathbf{y}$ is an **affine combination** of a set of points $X = \{\mathbf{x_0}, \ldots \mathbf{x_d}\}$ if there exist $t_i \in \mathbb{Q}$ such that $\mathbf{y} = \sum_{i=0}^{d} t_i\mathbf{x_i}$ and $\sum_{i=0}^{d} t_i = 1$. The set X is called **affinely independent** if no element of X is an affine combination of the other elements. The set of all affine combinations of X is called the set **affinely generated** by X.

Proof. Let $d = \dim P$. The affine span of P is generated by $d+1$ affinely independent points $X = \{\mathbf{x_0}, \ldots, \mathbf{x_d}\}$. Obviously, $X' = \{(\mathbf{x_0},1)^T, \ldots, (\mathbf{x_d},1)^T\}$ is affinely independent. Moreover, $\mathbf{0}$ is affinely independent from X': there exist no t_i, $0 \le i \le d$, such that $\mathbf{0} = \sum_i t_i(\mathbf{x_i},1)^T$ with $\sum_i t_i = 1$. We now prove that $X'' = X' \cup \{\mathbf{0}\}$ generates the affine span of HP. This will be the case if any point in HP can be represented as an affine combination of X''.

By definition, any point in HP can be written as $(\mathbf{x},h)^T$, with $\mathbf{x} \in hP$, and thus $h^{-1}\mathbf{x} \in P$. Since X affinely generates the affine span of P, there exist t_i, $0 \le i \le d$, such that $h^{-1}\mathbf{x} = \sum_i t_i(\mathbf{x_i},1)$ and $\sum_i t_i = 1$. So

$$
\begin{pmatrix} \mathbf{x} \\ h \end{pmatrix} = \sum_i ht_i \begin{pmatrix} \mathbf{x_i} \\ 1 \end{pmatrix} = \sum_i ht_i \begin{pmatrix} \mathbf{x_i} \\ 1 \end{pmatrix} + (1 - \sum_i ht_i)\mathbf{0}.
$$

Since $\sum_i ht_i + 1 - \sum_i ht_i = 1$, we conclude that $(\mathbf{x}, h)^T$ is an affine combination of X''.

It follows from the definition of dim that $\dim HP = \dim P + 1$. □

Proposition 4.5 *If V is a validity domain of Q, then HV is a validity domain of $\bar{H}Q$.*

Proof. By the previous lemmas, the k-dimensional faces of $\widehat{Q}$ are transformed into $k+1$-dimensional faces of $\widehat{\bar{H}Q}$. Because H commutes with projection onto the parameter space (lemma 4.2), and distributes over union and intersection (lemma 4.1), a validity domain V is transformed into HV. □

4.3 Degeneracy

We now prove the main contribution of this paper, namely the elimination of degenerate domains.

Proposition 4.6 *Let Q be a parametric polyhedron with k parameters. The validity domains of $\bar{H}Q$ are never degenerate.*

Proof. Let HV be a validity domain of $\bar{H}Q$. Since $\dim HV = k+1$, certainly some hyperrectangle R can be found in HV. Then hR is also in HV, for all $h \in \mathbb{Q}_0$, and hence a hyperrectangle of arbitrary integer size vector can be found in HV. □

4.4 Reconstructing the Ehrhart quasi-polynomials

Clauss's method can compute the Ehrhart quasi-polynomials of $\bar{H}Q$ for any parametric polyhedron Q. To find $\mathcal{E}(Q, V)$, given $\mathcal{E}(\bar{H}Q, HV)$, it suffices to evaluate $\mathcal{E}(\bar{H}Q, HV)$ partially in $h = 1$.

Proposition 4.7 *For any parametric polytope Q, validity domain V and $\mathbf{n} \in V$:*

$$\mathcal{E}(\bar{H}Q, HV)(\mathbf{n}, 1) = \mathcal{E}(Q, V)(\mathbf{n}).$$

Proof. For $\mathbf{n} \in V$, we have:

$$\begin{aligned} \mathcal{E}(\bar{H}Q, HV)(\mathbf{n}, 1) &= \# \bar{H}Q(\mathbf{n}, 1) \\ &= \# Q(\mathbf{n}) \\ &= \mathcal{E}(Q, V)(\mathbf{n}) \end{aligned}$$

□

Example 4.1 We resolve the degenerate domain from Example 3.3. First, the parametric polytope Q is transformed into $(\bar{H}Q)(M, N, h) = \{(i, j)^T | M \le i \le N + h \text{ and } N \le j \le i + 2h\}$. The validity domains are: $HV_1 = \{(M, N, h)^T | M - N \le 2h\}$ and $HV_2 = \{(M, N, h)^T | -2h \le M - N \le h\}$. With Clauss's algorithm, we compute

$$\mathcal{E}(\bar{H}Q, HV_2)(N, M, h) = -\frac{1}{2}M^2 + (N - 2h - \frac{1}{2})M - \frac{1}{2}N^2 + (2h + \frac{1}{2})N + \frac{5}{2}h^2 + \frac{7}{2}h + 1.$$

Evaluating this in $h = 1$, we get

$$\mathcal{E}(\bar{H}Q, HV_2)(N, M, 1) = -\frac{1}{2}M^2 + (N - \frac{5}{2})M - \frac{1}{2}N^2 + \frac{5}{2}N + 7,$$

which is $\mathcal{E}(Q, V_2)(N, M)$.

5 Complexity analysis

In each recursion step, Clauss's method uses Gauss elimination to solve at most p systems of $d+1$ equations, where $p = \max_i p_i$. So each step has complexity $O(pd^3)$. For each step, at most $(d+1)p$ recursive calls are made, and the total recursion depth is $k-1$. In the base case, executed at most $(d+1)^k p^k$ times, volumes have to be computed which has complexity $O(vol)$. It follows that the total complexity is $O(p^k d^k (d^3 + vol))$.

The complexity of the computation of the volume of a non-parametric polyhedron depends on the used algorithm. A short overview is given in Section 6

The introduction of an additional parameter imposes no overhead in the case where the original algorithm can compute the Ehrhart quasi-polynomial: the extra parameter is only added when a degenerate domain is detected. Since there is no need to recompute the vertices and the validity domains after addition of the parameter, and the only characteristic of the problem that has changed is the number of parameters k, the complexity of the extended method is $O(p^{k+1} d^{k+1} (d^3 + vol))$.

6 Counting points in non-parametric polytopes

For counting points in a non-parametric polytope, there exists a simple enumeration algorithm. The polytope is intersected with hyperplanes perpendicular to one of the coordinate axes. Then the number of points in the original polytope is the sum of the number of points in the intersections, which is calculated recursively. In the one-dimensional case a polytope is just an interval $[a, b]$, which has $b - a + 1$ integer points.

In [BP99], Barvinok et al. describe an algorithm to compute the number of points in a polytope based on generating functions. Briefly, the algorithm is as follows. For every vertex, a cone is constructed, which is further decomposed into *unimodular* cones. For each of these unimodular cones the corresponding generating function is easily computed. The generating function for the polytope is the sum of the generating functions of the cones. To compute the number of points in the polytope, it suffices to evaluate the generating function in $\mathbf{1} = (1, 1, \ldots, 1)$. The execution time is mainly determined by the number of vertices and the number of unimodular cones.

In [VSB$^+$04], Verdoolaege et al. extended Barvinok's algorithm to parametric polytopes, thus computing Ehrhart quasi-polynomials. To our knowledge, this and Clauss's method extended with the results from this paper, are the only implemented algorithms to compute the Ehrhart quasi-polynomials of a general parametric polytope. The execution times of both Barvinok's original algorithm and the parametric version depend on the number of (parametric) vertices. If this number is relatively large, Clauss's method might be the better choice. In this case, the problem remains to find a suitable algorithm to compute the number of points in a non-parametric polytope: enumeration becomes very time consuming as the dimension or the volume of the polytope gets larger.

There are several algorithms based on complex integration, via Cauchy's Residue Theorem, e.g. [LZ03, Bec00]. Although theoretically these methods may be more efficient than Barvinok's for polytopes with a large number of vertices, none of these seem to be implemented for arbitrary polytopes. Since it is not yet clear how to adapt these algorithms for the parametric case, they might be good candidates for the base case in Clauss's method.

7 Conclusion

We have extended Clauss's method to compute Ehrhart quasi-polynomials, making it work for every parametric polytope. Although our extension requires the computation of more coefficients than follows form Theorem 2.1, this does not have an significant impact on the complexity when the pseudo-period and the dimension are small. Moreover, Clauss's method is particularly well-suited for quick adaptation to specific families of parametric polytopes. Indeed, it suffices to use a special purpose algorithm to count the points in a non-parametric polytope in the base case. This also reduces the effort needed to make a counting algorithm support parametric polytopes.

References

[Bec00] M. Beck. Counting lattice points by means of the residue theorem. *Ramanujan Journal*, 4(3):299–310, 2000.

[Bey04] K. Beyls. *Software Methods to Improve Data Locality and Cache Behavior*. PhD thesis, Ghent University, June 2004.

[BP99] A. Barvinok and J. Pommersheim. An algorithmic theory of lattice points in polyhedra. *New Perspectives in Algebraic Combinatorics*, 38:99–147, 1999.

[CL98] Ph. Clauss and V. Loechner. Parametric analysis of polyhedral iteration spaces. *Journal of VLSI Signal Processing*, 19(2):179–194, July 1998.

[CPHL01] S. Chatterjee, E. Parker, Ph. J. Hanlon, and A. R. Lebeck. Exact analysis of the cache behavior of nested loops. In *Proceedings of the ACM SIGPLAN 2001 conference on Programming language design and implementation*, pages 286–297. ACM Press, 2001.

[Ehr67] E. Ehrhart. Sur un problème de géométrie diophantienne linéaire I. *Journal für die reine und angewandte Mathematik*, 226:1–29, 1967.

[Loe99] V. Loechner. Polylib: A library for manipulating parameterized polyhedra. Technical report, ICPS, Université Louis Pasteur de Strasbourg, France, March 1999.

[LW97] V. Loechner and D. K. Wilde. Parameterized polyhedra and their vertices. *International Journal of Parallel Programming*, 25(6):525–549, 1997.

[LZ03] J. B. Lasserre and E. S. Zeron. On counting integral points in a convex rational polytope. *Mathematics of Operations Research*, 28(4):853–870, 2003.

[Sch86] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, Chichester, 1986.

[SVBL04] R. Seghir, S. Verdoolaege, K. Beyls, and V. Loechner. Analytical computation of Ehrhart polynomials and its application in compile-time generated cache hints. Technical report, Strasbourg, February 2004.

[TKD02] A. Turjan, B. Kienhuis, and E. Deprettere. A compile time based approach for solving out-of-order communication in Kahn Process Networks. In *Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures, and Processors*. IEEE Computer Society, 2002.

[VSB+04] S. Verdoolaege, R. Seghir, K. Beyls, V. Loechner, and M. Bruynooghe. Analytical computation of Ehrhart polynomials: Enabling more compiler analyses and optimizations. In *Proceedings of the 2004 international conference on Compilers, architecture, and synthesis for embedded systems*, pages 248–258. ACM Press, 2004.