# Uniform Random Sampling in Polyhedra

Benoît Meister, Philippe Clauss

Reservoir Labs, INRIA CAMUS - University of Strasbourg

IMPACT 2020, 22 Jan 2020

# Outline

# Refresher
## Parametric polyhedron

Polyhedron : convex set delimited by affine (linear + constant) constraints.
Non-parametric :

$$P(x) : Ax + b \geq 0, x \in V \tag{1}$$

Parametric : treat some variables as symbolic constants

$$P(x, N) : Ax + BN + c \geq 0, (x, N) \in V \tag{2}$$

Example : $n \times m$ box in $2 - d$ space

$$Q(i, j, n, m) = \{(i, j) \in \mathbb{Z}^2 : 0 \leq i \leq n; 0 \leq j \leq m\} \tag{3}$$

# Problem

Sample integer points uniformly in a bounded parametric polyhedron

- Each point has the same probability of being sampled
  - Often, there is no (other) known prior
  - Desirable property of a random search
  - Simple to understand
- Integer points
  - Generalizes trivially to any lattice, even real
- Parametric polyhedron
  - Compute sampling function once, use it for any value of the parameters

# Use cases

## 1- Iterative compilation

- Look for a schedule using a search
- Without any known prior

1. Bastoul's method [Bastoul et al. 05, 16]
    - Sample a hyper-rectangular superset of $P$
    - Pull it back into $P$
    - Minimizes distance w/ sampled schedule
    - No obvious way to get uniform distribution
    - Useful if the assumption is that solutions lie around the faces
    - (Still not uniform within set of edge solutions)

# Use cases

## 1- Iterative compilation

2. Pouchet's method [Pouchet et al. 07]
    - Generate scanning loops of $P$
    - Draw $d = [1, \#P]$ and stop when $d$ points scanned
    - Can adapt to uniform distribution by re-scanning every time
    - $O(n\#P)$ to draw $n$ samples.

- Looking for a method that has a $O(n)$ cost

# Use cases

## 2- Stochastic Polyhedral Operators

Polyhedral model works with parametric polyhedra

- Don't know what their value will be in practice
- Automatic parallelization is about making choices
- Need to compare, relate, sort, etc. : relationship $rel(A, B)$
- Problem when result is itself parametric : $rel(A, B, N)$
  - Valuable to know if relationship is *mostly* true
  - Or to which amount (probability) it is true

# Use cases

## 2- Stochastic Polyhedral Operators

Naive method sometimes available. Let $C(N)$ be the domain of $rel(A, B, N)$.

- Form the parameter sub-domain $T$ of $C$.
- Probability $= \frac{\#T}{\#C}$
- Issues
  - Can't always compute $T$
  - Counting can be expensive

# Use cases

## 2- Stochastic Polyhedral Operators

Naive method sometimes available. Let $C(N)$ be the domain of $rel(A, B, N)$.

- Form the parameter sub-domain $T$ of $C$.
- Probability $= \frac{\#T}{\#C}$
- Issues
  - Can't always compute $T$
    - E.g., compare $\#A(N)$ and $\#B(N)$
  - Counting can be expensive

# Use cases

## 2- Stochastic Polyhedral Operators

- Compute a stochastic approximation of the relationship
- Sample (uniformly) : $E(rel, A, B) = \frac{1}{|U|} \sum_{N \in U} rel(A, B, N)$
  - We can always compute $rel(A, B, N)$
  - Doesn't need polyhedral counting
  - Can trade off speed for precision (# samples)

# Use cases
## 3- Random testing

Some uses :

- Test programs where affine bounds are known on inputs
  - Bounds can come from static or dynamic analysis, or both
- Generate random inputs for polyhedral libraries
  - Generate parametric polyhedra
  - Instantiate (some) parameters from sampling
  - Explores size
- Generate random polyhedral programs to test a compiler
  - R-Stream's nightly tests include this

# Method

## General Idea

- Get a known bijective mapping from integer points of $P$ to $\mathbb{N}$ using *ranking functions*
- Invert the mapping - the trahrhe method to get $invrank_P(I, N)$
- sample uniformly $s \in [1, \#P(N)]$ and compute $X = invrank_P(s, )$
- since $invrank_P(s, N)$ is a bijection, $X$ is sampled uniformly in $P$

# Preliminaries

## Ehrhart Polynomials [Clauss 96, Clauss & Loechner 98]

- Integer-valued polynomials
- Express the exact number of integer points contained in a polytope which depends linearly on integer parameters
- For a $d$-dimensional polytope depending on parameters $p_1, p_2, \ldots, p_m$ :
  Polynomial of degree $d$ whose variables are $p_1, p_2, \ldots, p_m$, and whose coefficients are periodic numbers
- Can be automatically computed using existing algorithm implementations as the one of the barvinok library

# Preliminaries

## Ranking Polynomials [Clauss & Meister 2000]

```
pc=0;
/* A general affine loop nest */
for(i_0=l_0; i_0<h_0; i_0++)
 for(i_1=l_1(i_0); i_1<h_1(i_0); i_1++)
  ...
   for(i_d=l_d(i_0,i_1,...,i_{d-1}); i_d<h_d(i_0,i_1,...,i_{d-1}); i_d++)
   {
    pc++;
    if ((i_0==j_0) && (i_1==j_1) && ... && (i_d==j_d))
     printf("%d\n",pc);
   }
```

$$\Updownarrow$$

```
printf("%d\n", RankingPolynomial(j_0,j_1,...,j_{d-1}));
```

# Preliminaries

## Ranking Polynomial : example

$$P = \{(i,j,k) \in \mathbb{Z}^3 | 0 \leq i < N, 0 \leq j \leq i, 0 \leq k < M\}$$

Rank of $(i_0, j_0, k_0) \in P$

= number of points that are lexicographically less than $(i_0, j_0, k_0)$ (included) :

$$\boxed{\begin{aligned} Rank(i_0, j_0, k_0) = \#\{(i,j,k) \mid (i,j,k) \trianglelefteq (i_0, j_0, k_0), \\ 0 \leq i < N, 0 \leq j \leq i, 0 \leq k < M\} \end{aligned}}$$

where $\trianglelefteq$ denotes the lexicographic order

# Preliminaries

## Ranking Polynomial : example

$$
\begin{aligned}
(i,j,k) \trianglelefteq (i_0,j_0,k_0) \Leftrightarrow & (i < i_0) \text{ or } (i = i_0 \text{ and } j < j_0) \text{ or} \\
& (i = i_0 \text{ and } j = j_0 \text{ and } k \leq k_0)
\end{aligned}
$$

$\Rightarrow Rank(i_0, j_0, k_0)$ is the sum of 3 Ehrhart polynomials :

$$
\begin{aligned}
Rank(i_0,j_0,k_0) &= \#\{(i,j,k) \mid 0 \leq i < i_0, 0 \leq j \leq i, 0 \leq k < M\} \\
&+ \#\{(i,j,k) \mid i = i_0, 0 \leq j < j_0, 0 \leq k < M\} \\
&+ \#\{(i,j,k) \mid i = i_0, j = j_0, 0 \leq k \leq k_0\} \\
&= \frac{M\, i_0\, (i_0+1)}{2} + M\, j_0 + k_0 + 1 \\
&= \boxed{\frac{2\, k_0 + 2\, M\, j_0 + M\, i_0^2 + M\, i_0 + 2}{2}}
\end{aligned}
$$

# Preliminaries

## Ranking Polynomial

Properties of Ranking Polynomials :

- monotonically increasing over the integers, from 1 to the total number of points, relatively to the lexicographic order of the tuples
- define a bijection between the tuples and the interval of successive integers, between 1 and the total number of points

# Trahrhe Expressions

*Ranking Polynomial* :
Affine loop indices tuple $\rightarrow$ Rank

*Trahrhe Expressions* :
Rank $\rightarrow$ Affine loop indices tuple

$$\textit{Trahrhe Expressions} = \textit{Ranking Polynomial}^{-1}$$

# Trahrhe Expressions

## Example

$$P = \{(i,j,k) \in \mathbb{Z}^3 | 0 \leq i < N, 0 \leq j \leq i, 0 \leq k < M\}$$

1. $Rank(i,j,k) = \frac{2\,k+2\,M\,j+M\,i^2+M\,i+2}{2}$
2. Solve $Rank(s,0,0) - pc = \frac{M\,s^2+M\,s+2}{2} - pc = 0$. This equation has two solutions :

$$s_1 = -\frac{\sqrt{8\,M\,pc + M^2 - 8\,M} + M}{2\,M}, s_2 = \frac{\sqrt{8\,M\,pc + M^2 - 8\,M} - M}{2\,M}$$

When $pc = 1$, $s_2 = 0$. Thus $t_1 = \left\lfloor \frac{\sqrt{8\,M\,pc+M^2-8\,M}-M}{2\,M} \right\rfloor$

# Trahrhe Expressions
## Example

1. Solve $Rank(t_1, s, 0) - pc = \frac{M\,t_1^2 + M\,t_1 + 2\,M\,s + 2}{2} - pc = 0$. This equation has one solution. Thus $t_2 = \left\lfloor -\frac{M\,t_1^2 + M\,t_1 - 2\,pc + 2}{2\,M} \right\rfloor$
2. $t_3 = pc - Rank(t_1, t_2, 0) = -\frac{2\,M\,t_2 + M\,t_1^2 + M\,t_1 - 2\,pc + 2}{2}$

$$Trahrhe(pc) = \left( \left\lfloor \frac{\sqrt{8\,M\,pc + M^2 - 8\,M} - M}{2\,M} \right\rfloor, \left\lfloor -\frac{M\,t_1^2 + M\,t_1 - 2\,pc + 2}{2\,M} \right\rfloor, -\frac{2\,M\,t_2 + M\,t_1^2 + M\,t_1 - 2\,pc + 2}{2} \right)$$

- Closed-form bijection $invrank_P$ from $\mathbb{N}$ to the integer points of parametric polyhedron $P$
- Uniform sampling method in $O(n)$ for $n$ samples
- Stochastic polyhedral relationships
  - Tell us more about parametric polyhedra at compile time
  - Have a built-in performance-precision tradeoff ($n$)
  - Only cost $n$ times the cost of the non-parametric relationship
- Ranking is defined for a lattice basis
  - Many possible rankings, i.e., sampling functions
  - Limited form of unbounded domains are supported