Parametric Integer Hull

Marc Moreno Maza   Linxiao Wang

ORCCA, University of Western Ontario, Canada

November 06, 2020

Outline

1. Why
2. What
3. How (On-going work)

Outline

1. Why
2. What
3. How (On-going work)

The Delinearization Problem

A simple example

Input:

$$
\begin{array}{l}
(\mathbf{i} = 0;\ \mathbf{i} < n;\ \mathbf{i}{+}{+}) \\
\quad (\mathbf{j} = 0; \mathbf{j} < m;\ \mathbf{j}{+}{+}) \\
\quad A[i \times m + j] \leftarrow \cdots \\
\quad \ldots
\end{array}
$$

Output:

$$
\begin{array}{l}
(\mathbf{i} = 0;\ \mathbf{i} < n;\ \mathbf{i}{+}{+}) \\
\quad (\mathbf{j} = 0; \mathbf{j} < m;\ \mathbf{j}{+}{+}) \\
\quad \tilde{A}[i][j] \leftarrow \cdots \\
\quad \ldots
\end{array}
$$

- Data dependency analysis:
  ex. Are references $A(\mathbf{i}_1 + 10\,\mathbf{j}_1)$ and $A(\mathbf{i}_2 + 5)$, $0 \le \mathbf{i}_1, \mathbf{i}_2 \le n$, $0 \le \mathbf{j}_1, \mathbf{j}_2 \le m$ independent?
- Loop parallelization
- Loop optimization such as blocking

Input:

$(\mathbf{i}_1 \cdots ;\ \cdots ;\ \mathbf{i}_1{+}{+})$

$\ldots (\mathbf{i}_d \cdots ; \cdots ;\ \mathbf{i}_d{+}{+})$

$A[R(\mathbf{i}_1, \ldots, \mathbf{i}_d, \mathbf{m}_1, \ldots, \mathbf{m}_\delta)] \leftarrow \cdots$

$\ldots$

- $\mathbf{i}_1, \ldots, \mathbf{i}_d$ take non-negative integer values such that
$$L \begin{pmatrix} \mathbf{i}_1 \\ \vdots \\ \mathbf{i}_d \end{pmatrix} \leq \begin{pmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_d \end{pmatrix},$$
- L is a lower-triangular full-rank matrix over $\mathbb{Z}$ (known at compile time) defining the iteration domain
- $\mathbf{m}_1, \ldots, \mathbf{m}_\delta, \mathbf{r}_1, \ldots, \mathbf{r}_\delta$: data parameters (known only at execution time)
- $R(\mathbf{i}_1, \ldots, \mathbf{i}_d, \mathbf{m}_1, \ldots, \mathbf{m}_\delta)$ is a polynomial, the coefficients of which are known at compile time.

Input:

$(\mathbf{i}_1 \cdots ;\ \cdots ;\ \mathbf{i}_1{+}{+})$
$\ldots (\mathbf{i}_d \cdots ; \cdots ;\ \mathbf{i}_d{+}{+})$
$A[R(\mathbf{i}_1, \ldots, \mathbf{i}_d, \mathbf{m}_1, \ldots, \mathbf{m}_\delta)] \leftarrow \cdots$
$\cdots$

- $\mathbf{i}_1, \ldots, \mathbf{i}_d$ take non-negative integer values such that
$$L \begin{pmatrix} \mathbf{i}_1 \\ \vdots \\ \mathbf{i}_d \end{pmatrix} \leq \begin{pmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_d \end{pmatrix},$$
- L is a lower-triangular full-rank matrix over $\mathbb{Z}$ (known at compile time) defining the iteration domain
- $\mathbf{m}_1, \ldots, \mathbf{m}_\delta, \mathbf{r}_1, \ldots, \mathbf{r}_\delta$: data parameters (known only at execution time)
- $R(\mathbf{i}_1, \ldots, \mathbf{i}_d, \mathbf{m}_1, \ldots, \mathbf{m}_\delta)$ is a polynomial, the coefficients of which are known at compile time.

Output:

$(\mathbf{i}_1 \cdots ;\ \cdots ;\ \mathbf{i}_1{+}{+})$
$\ldots (\mathbf{i}_d \cdots ; \cdots ;\ \mathbf{i}_d{+}{+})$
$\tilde{A}[f_1] \cdots [f_\delta] \leftarrow \cdots\ \ldots$

- $f_1, \ldots, f_\delta$ are affine forms in $\mathbf{i}_1, \ldots, \mathbf{i}_d$ the coefficients of which are integers to-be-determined,
- $\tilde{A}$ is an $\mathbf{M}_1 \times \cdots \times \mathbf{M}_\delta$-array,
- $\mathbf{M}_1, \ldots, \mathbf{M}_\delta$ are affine forms in $\mathbf{m}_1, \ldots, \mathbf{m}_d$ the coefficients of which are integers TBD,

such that:

$$\boxed{R = f_1 \mathbf{M}_2 \cdots \mathbf{M}_\delta \ + \ \cdots \ + \ f_{\delta-1} \mathbf{M}_2 + f}$$

holds and for each $(\mathbf{i}_1, \ldots, \mathbf{i}_d)$ in the iteration domain we have:

$$\boxed{0 \leq f_1 < \mathbf{M}_1, \ \ldots, 0 \leq f_\delta < \mathbf{M}_\delta.}$$

The sub-problems

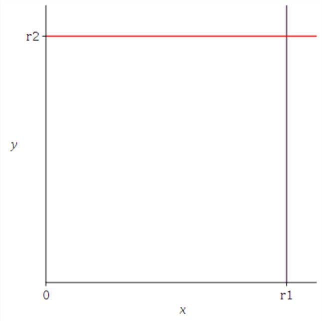
Polynomial system solving

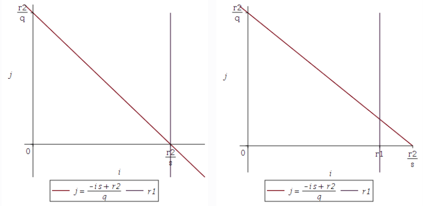
1. Expressing the coefficients of $f_1, \ldots, f_\delta$ and $\mathbf{M}_1, \ldots, \mathbf{M}_\delta$ as functions of the coefficients of R
2. This can be done off-line (that is, before compile-time) once d and δ are fixed.
3. Recall that the matrix L and the coefficients of the polynomial R are integer values known at compile-time.

Quantifier elimination

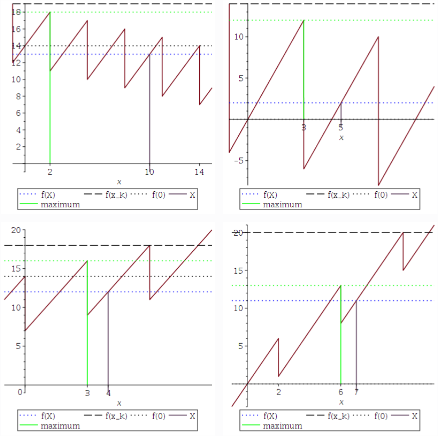
1. The constraint: for each $(\mathbf{i}_1, \ldots, \mathbf{i}_d)$ in the iteration domain we have:
$$0 \leq f_1 < \mathbf{M}_1, \quad \ldots, 0 \leq f_\delta < \mathbf{M}_\delta$$
implies constraints on the coefficients of $f_1, \ldots, f_\delta$. and $\mathbf{M}_1, \ldots, \mathbf{M}_\delta$
2. Off-line, this is a non-linear QE problem which can only be solved over the reals (not over the integers). The obtained constraint is then sufficient but not necessary.
3. At compile time, the coefficients of L and R are known and the QE problem can be reduced to Presburger arithmetic (that is, QE on affine forms over $\mathbb{Z}$) which can be solved by software like ISL.
4. This is the point of view of the paper *Optimistic Delinearization of Parametrically Sized Arrays* by T. Grosser, J. Ramanujam, L.-N. Pouchet, P. Sadayappan and S. Pop (ICS15).
5. At run-time, $\mathbf{m}_1, \ldots, \mathbf{m}_\delta, \mathbf{r}_1, \ldots, \mathbf{r}_\delta$ are known and the QE problem reduces to optimize peice-wise linear functions (actually *sawtooth functions*).

The iteration domain looks like:



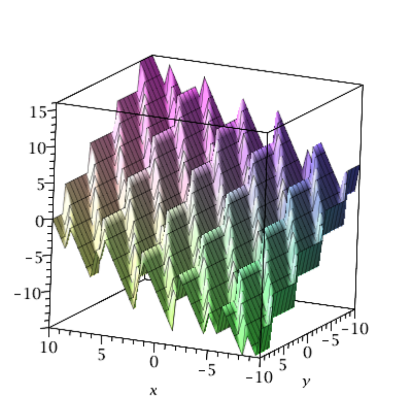


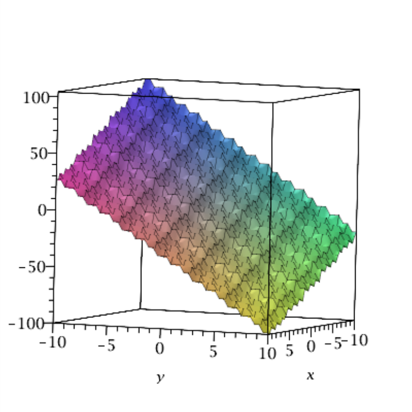
There are mainly four different cases. Based on the shape of the plots, we call this type of functions "sawtooth" functions.



The 3D Case QE solving

"sawtooth" functions in 3D case.





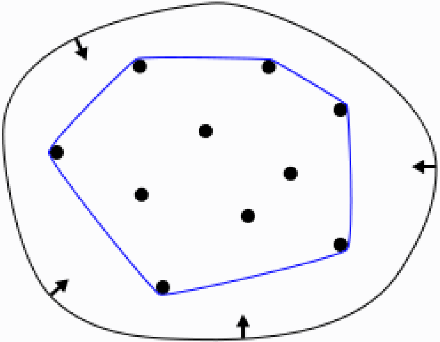
- Only integer points matter in loop iterations.
- The integer hull problem

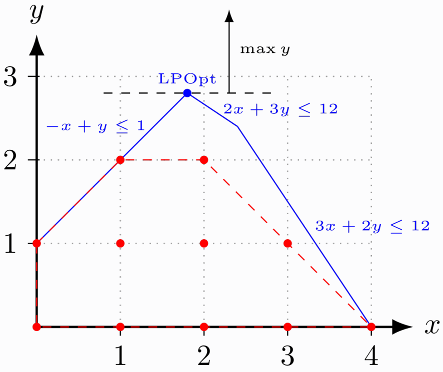
Outline

1. Why
2. What
3. How (On-going work)

Convex hull and Integer hull

- In geometry, the convex hull of a shape is the smallest convex set that contains it.
- Integer hull: when all the vertices are integer points



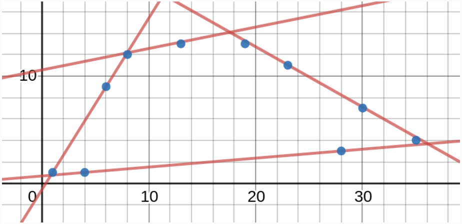


Outline

An example

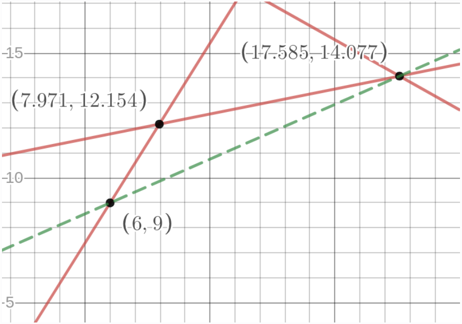
For a polygon given by the facets

$$\begin{cases} y & \leq & \frac{1}{5}\,x + \frac{264}{25} \\ y & \leq & -\frac{9}{16}\,x + \frac{767}{32} \\ y & \leq & \frac{8}{5}\,x - \frac{3}{5} \\ y & \geq & \frac{1}{12}\,x + \frac{2}{3} \end{cases} \tag{1}$$

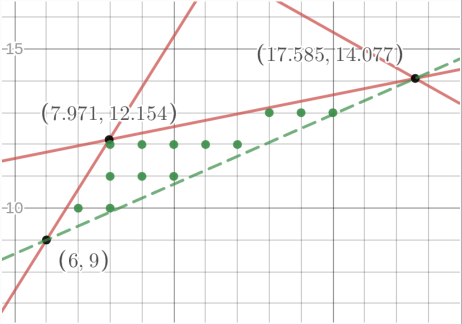


An example (cont.)

Consider one rational vertex:



Triangle Rasterization:



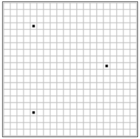


Figure 1

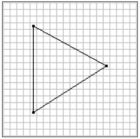


Figure 2

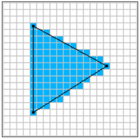
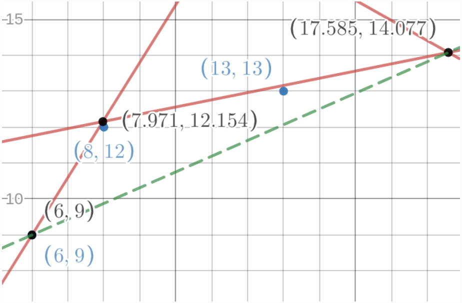


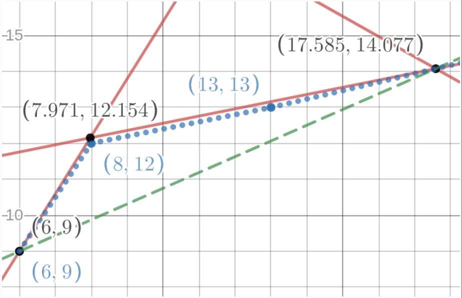
Figure 3

An example (cont.)

Find the convex hull of all the integer points:



Connect the new vertices:



The Parametric Case

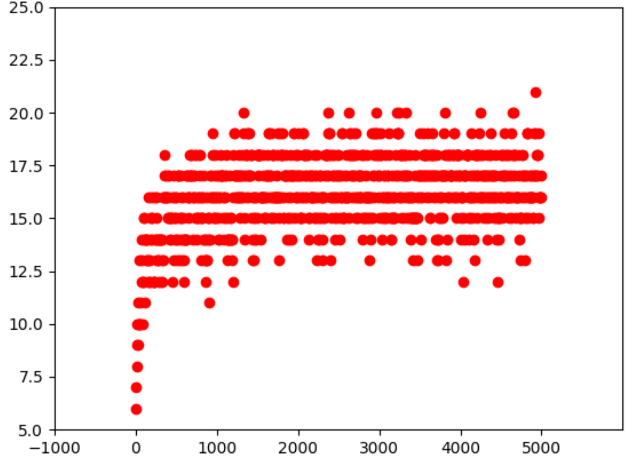
The triangle will be given by:

$$\begin{cases} b_1 & \leq & a_1\,x + y \\ b_2 & \leq & a_2\,x + y \\ b_3 & \leq & a_3\,x + y \end{cases} \tag{2}$$

- Our goal: generate in constant time a program for computing the integer hull, and evaluate that program when the value of parameters are given.
- Parametric code generation: branch at conditional statements
- Consider every integer points would cause too many branches. (At least linear complexity)
- Solution: Finding the integer hull of a triangle in constant time
- Cook, W., Hartmann, M., Kannan, R., & McDiarmid, C. (1992). On integer points in polyhedra. Combinatorica, 12(1), 27-37.
  Suggests that the number of points in an integer hull has a upper limit related to the size of the polygon
- But is there a limit for a fixed shape?

Periodic behavior of the integer hulls of a fix-shaped triangle

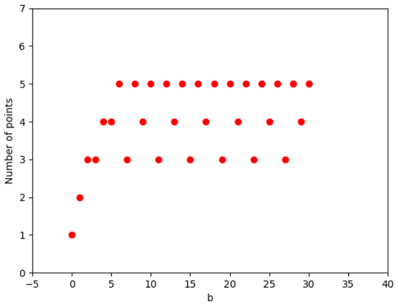
Number of points of a complex example:



Periodic behavior of the integer hulls of a fix-shaped triangle

Consider the triangle defined by the facets

$$\begin{cases} y & \leq & 2\,x \\ y & \geq & 0 \\ y & \leq & -2\,x + b \end{cases} \tag{3}$$

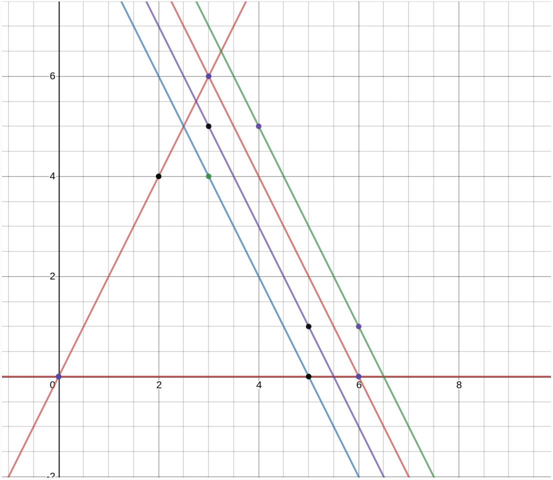


b	n	x	y
$4i-2$	4	$[2i-1, i, i-1, 0]$	$[0, 2i-2, 2i-2, 0]$
$4i-1$	5	$[2i-1, 2i-1, i, i-1, 0]$	$[0, 1, 2i-1, 2i-2, 0]$
$4i$	3	$[2i, i, 0]$	$[0, 2i, 0]$
$4i+1$	5	$[2i, 2i, i+1, i, 0]$	$[0, 1, 2i-1, 2i, 0]$

Table: Points in the integer hull

Periodic behavior of the integer hulls of a fix-shaped triangle

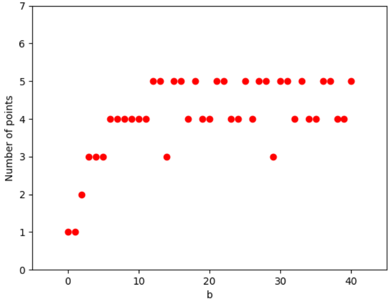
Visulization of the above example



Periodic behavior of the integer hulls of a fix-shaped triangle

Consider the triangle defined by the facets

$$\left\{ \begin{array}{lll} y & = & 2\,x \\ y & = & 0 \\ y & = & -3\,x + b \end{array} \right. \qquad (4)$$



This example has a period of 15 and a inner period of 3

$b - 15\,i$	$b - 3\,j$	n	x	y
0	0	3	$[j, 3i, 0]$	$[0, 6i, 0]$
1	1	5	$[j, j, 3i+1, 3i, 0]$	$[0, 1, 6i-2, 6i, 0]$
2	2	3	$[j, j, 3i+1, 3i, 0]$	$[0, 2, 6i-1, 6i, 0]$
3	0	4	$[j, 3i+1, 3i, 0]$	$[0, 6i, 6i, 0]$
4	1	5	$[j, j, 3i+1, 3i, 0]$	$[0, 1, 6i+1, 6i, 0]$
5	2	4	$[j, j, 3i+1, 0]$	$[0, 2, 6i+2, 0]$
6	0	4	$[j, 3i+2, 3i+1, 0]$	$[0, 6i, 6i+2, 0]$
7	1	5	$[j, j, 3i+2, 3i+1, 0]$	$[0, 1, 6i+1, 6i+2, 0]$
8	2	5	$[j, j, 3i+2, 3i+1, 0]$	$[0, 2, 6i+2, 6i+2, 0]$
9	0	4	$[j, 3i+2, 3i+1, 0]$	$[0, 6i+3, 6i+2, 0]$
10	1	4	$[j, j, 3i+2, 0]$	$[0, 1, 6i+4, 0]$
11	2	5	$[j, j, 3i+3, 3i+2, 0]$	$[0, 2, 6i+2, 6i+4, 0]$
12	0	4	$[j, 3i+3, 3i+2, 0]$	$[0, 6i+3, 6i+4, 0]$
13	1	5	$[j, j, 3i+3, 3i+2, 0]$	$[0, 1, 6i+4, 6i+4, 0]$
14	2	5	$[j, j, 3i+3, 3i+2, 0]$	$[0, 2, 6i+5, 6i+4, 0]$

Table: Example with a double period

Periodic behavior of the integer hulls of a fix-shaped triangle

Theorem

For a triangle given by the facets

$$\left\{ \begin{array}{lcl} y & = & a_1\, x + b_1 \\ y & = & a_2\, x + b_2 \\ y & = & a_3\, x + b_3 \end{array} \right. \tag{5}$$

where $a_i = \frac{m_i}{d_i}$ are rational numbers, m_i, d_i are integer numbers. The period of the integer hull is given by

$$LCM(|m_1 d_2 d_3 - m_2 d_1 d_3|, |m_1 d_2 d_3 - m_3 d_1 d_3|, |m_2 d_1 d_3 - m_3 d_1 d_2|) \tag{6}$$

The vertices are:

$$\left\{ \begin{array}{lcl} x_1 & = & \frac{b_1 - b_2}{a_1 - a_2} = \frac{b_1 - b_2}{\frac{m_1 d_2 - m_2 d_1}{d_1 d_2}} = \frac{d_1 d_2 (b_1 - b_2)}{m_1 d_2 - m_2 d_1} = \frac{d_1 d_2 d_3 (b_1 - b_2)}{m_1 d_2 d_3 - m_2 d_1 d_3} \\ x_2 & = & \frac{b_1 - b_3}{a_1 - a_3} = \frac{b_1 - b_3}{\frac{m_1 d_3 - m_3 d_1}{d_1 d_3}} = \frac{d_1 d_3 (b_1 - b_3)}{m_1 d_3 - m_3 d_1} = \frac{d_1 d_2 d_3 (b_1 - b_3)}{m_1 d_2 d_3 - m_3 d_1 d_2} \\ x_3 & = & \frac{b_2 - b_3}{a_2 - a_3} = \frac{b_2 - b_3}{\frac{m_2 d_3 - m_3 d_2}{d_3 d_3}} = \frac{d_2 d_3 (b_2 - b_3)}{m_2 d_3 - m_3 d_2} = \frac{d_1 d_2 d_3 (b_2 - b_3)}{m_2 d_1 d_3 - m_3 d_1 d_2} \end{array} \right. \tag{7}$$

The period is the LCM of the denominators of the x_i.

On-going work

What's next?

- Formally prove the above theorem
- General form of the points in a cycle (like the tables above)
- Expand the results to other shapes and higher dimensions
- Parametric integer hull
- Delinearization

Thank You!

Your Questions?