Computing the integer points of polyhedral sets with the Polyhedra library

1. Introduction

The Polyhedra library offers commands to compute the integer solutions to a linear system. The input system contains linear equations, as well as linear inequalities. For a given linear system, the coefficients must be in the field of rational numbers.

Let us instanciate the library for working with 4-dimensional polyhedral sets with coordinates x, y, z, w :

```
> restart:
  PLHD := Polyhedra([x,y,z,w]);with(PLHD);
```

$$PLHD := \textbf{module}()\ \ldots\ \textbf{end module}$$

$$[Canonicalize, DarkShadow, Display, Equations, GreyShadow, Inequalities, IntegerSolve, Ranking, RealShadow, UnfoldMaxExpressions, UnfoldMinExpressions] \quad (1.1)$$

2. Functions

This section introduces the user to the main command of the Polyhedra library.

2.1 Canonicalize

The input system consists of equations, inequalities and the boolean value HNF with default value true. The command finds the integer solutions common tothe input equations, as well to the implicit equations in the inequality system. If HNF is set to be true, this function will use the HNF method, otherwise, it will use the Pugh's method. Moreover, after replacing the solved equations with their solutions, the inequality system is simplified to a triangularized form, which has no implicit equations and no redundant inequalities. Each inequality in this triangularized fo m has content 1. Finally, this function will output the solved equations and the simplified inequalities.

```
> equations := [];
  inequalities := [ 2 * x+3 * y-4 * z+3 * w <=  1,
         -2 * x-3 * y+4 * z-3 * w <=  -1,
         -13 * x-18 * y+24 * z-20 * w  <=  -1,
         -26 * x-40 * y+54 * z-39 * w  <=  0,
         -24 * x-38 * y+49 * z-31 * w  <=  5,
         54 * x+81 * y-109 * z+81 * w  <=  2];
```

$$equations := [\,]$$

$$inequalities := [2\,x+3\,y-4\,z+3\,w \le 1,\ -2\,x-3\,y+4\,z-3\,w \le -1,\ -13\,x-18\,y+24\,z-20\,w \le -1,\ -26\,x-40\,y+54\,z-39\,w \le 0,\ -24\,x-38\,y+49\,z-31\,w \le 5,\ 54\,x+81\,y-109\,z+81\,w \le 2] \quad (2.1.1)$$

```
> canon := Canonicalize(equations, inequalities, [x, y, z, w],
```

```
  true);
```

$$canon := [[w=1-2\,t\_1-t\_2,\ x=-1+3\,t\_1+2\,t\_3,\ y=t\_2,\ z=t\_3],\ [-t\_3 \le -25,\ t\_3 \le 34,\ 2\,t\_3-t\_2 \le 13,\ 13\,t\_2-19\,t\_3 \le 72,\ 3\,t\_2-4\,t\_3 \le 29,\ t\_3-10\,t\_1-7\,t\_2 \le 12,\ -2\,t\_1-t\_2 \le 17,\ 2\,t\_2-2\,t\_3+t\_1 \le 6],\ [x, y, z, w, t\_1, t\_2, t\_3]] \quad (2.1.2)$$

```
> canon := Canonicalize(equations, inequalities, [x, y, z, w],
  false);
```

$$canon := [[y=-2\,t\_0+2\,z-w+1,\ x=3\,t\_0-z-1],\ [-z \le -25,\ z \le 34,\ 2\,z-w \le 50,\ 13\,w-5\,z \le 73,\ w \le 18,\ z-2\,w-3\,t\_0 \le 4,\ 2\,t\_0+w \le 14,\ -3\,z+7\,w+4\,t\_0 \le 19],\ [x, y, t\_0, w, z]] \quad (2.1.3)$$

2.2 DarkShadow

The input system consist of an inequality system, a variable list specifying an elimination order as well boolean values HNF, partition and recursive, all with default values true. This function aims at giving a representation of the dark shadow of the polyhedron defined by the inequality system w.r.t. the first variable in the variable list. This representation is a system of linear inequalities. If recursive is set to be true, the function will call IntegerSolve on this representation with dark_shadow and grey_shadow both equal to true, and HNF and partition set as the input of the original call. If recursive is set to false, this representation is simply returned.

```
> equations := []; inequalities := [3*x-2*y+z<= 7, -2*x+2*y-z
  <= 12, -4*x+y+3*z <= 15, -y <= -25];
```

$$equations := [\ ]$$

$$inequalities := [3\,x-2\,y+z \le 7,\ -2\,x+2\,y-z \le 12,\ -4\,x+y+3\,z \le 15,\ -y \le -25] \quad (2.2.1)$$

```
> DS := DarkShadow(inequalities, [x,y,z]) ;
  map(Display, DS);
```

$$DS := [Z\_polyhedron]$$

$$\left[\left\{\begin{array}{c} -y \le -25 \\ -5\,y+13\,z \le 67 \\ 2\,y-z \le 48 \\ -z \le -2 \\ z \le 17 \end{array}\right.\right] \quad (2.2.2)$$

2.3 GreyShadow

The input system consists of the inequality system, a variable list specifying an elimination order and the boolean values HNF, partition and cleanup, all with

default values true.
This function computes the grey shadow w.r.t. the first variable in the variable list. Each part of the output system consists of solved equations and inequalities. If HNF is set to be true, the solved equations is obtained by HNF method, otherwise, by Pugh's method. If partition is set to be true, each integer point in the grey shadow must be in exactly one part of the output. The boolean value recursive is similar to what we have introduced in Section 2.2.

```
> GS := GreyShadow(inequalities, [x,y,z]);
  map(Display, GS);
```

$$GS := [Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron]$$

$$\left[\begin{cases} x = 15 \\ y = 27 \\ z = 16 \end{cases}, \begin{cases} x = 18 \\ y = 33 \\ z = 18 \end{cases}, \begin{cases} x = 14 \\ y = 25 \\ z = 15 \end{cases}, \begin{cases} x = 19 \\ y = 50 + t\_2 \\ z = 50 + 2\, t\_2 \\ -t\_2 \le 25 \\ t\_2 \le -16 \end{cases}\right] \tag{2.3.1}$$

2.4 RealShadow

This function mainly uses the command RegularChains:-SemiAlgebraicSetTools:-LinearSolve to obtaina a representation of the input linear system. Note that simplification is done over the rational numbers whereas Canonicalize, DarkShadow,
GreyShadow and IntegerSolve return integer points.

```
> RS := RealShadow(equations, inequalities, [x, y, z]);
```

$$RS := \left[0 \le z, z \le \frac{132}{7}, 25 \le y, -\frac{73}{5} + \frac{13}{5} z \le y, y \le 25 + \frac{1}{2} z\right] \tag{2.4.1}$$

2.5 Plot

This section illustrates the other one by ploting an input polyhedron, its dark shadow, its real shdow and the difference between the latter two.

```
> with(PolyhedralSets):
  sys := [op(equations), op(inequalities)];
  SP := PolyhedralSet(sys);
  DP := PolyhedralSet([op(DS[1][EQS]), op(DS[1][INEQS]), x=0],
  [x,y,z]);
  RP := PolyhedralSet([op(RS), x=0], [x,y,z]);
  point_in_DP := PolyhedralSet([x=0, y=29, z=9]);
  Region_in_sys := PolyhedralSet([op(sys), y=29, z=9]);
  Plot([SP, DP, RP, point_in_DP, Region_in_sys], color=[blue,
  blue, grey, red, red], transparency = [0.8, 0.5, 0.8, 0.5,
  0.5]);
```

$$sys := [3\,x - 2\,y + z \le 7, -2\,x + 2\,y - z \le 12, -4\,x + y + 3\,z \le 15, -y \le -25]$$

$$SP := \begin{cases} Coordinates & : & [x, y, z] \\ Relations & : & \left[-y \le -25,\ -x + \frac{y}{4} + \frac{3\,z}{4} \le \frac{15}{4},\ -x + y - \frac{z}{2} \le 6,\ x - \frac{2\,y}{3} + \frac{z}{3} \le \right. \end{cases}$$

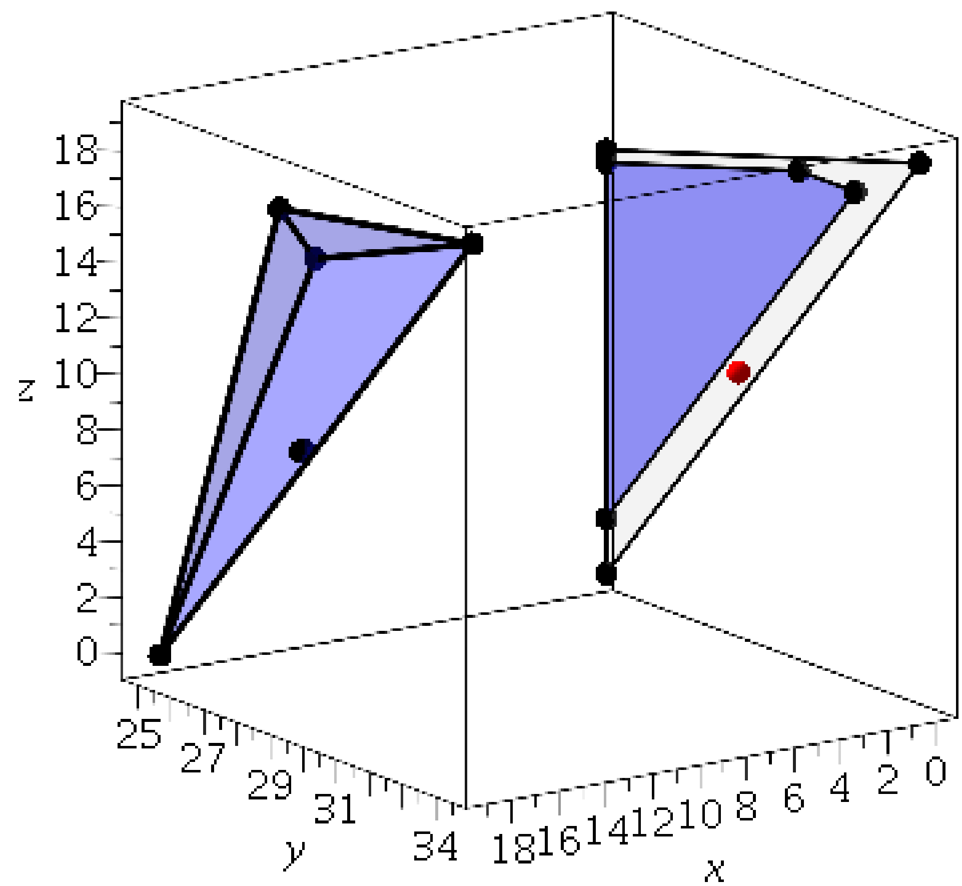
$$DP :=$$

$$\begin{cases} Coordinates & : & [x, y, z] \\ Relations & : & \left[z \le 17,\ -y \le -25,\ -y + \frac{13\,z}{5} \le \frac{67}{5},\ y - \frac{z}{2} \le 24,\ x = 0\right] \end{cases}$$

$$RP := \begin{cases} Coordinates & : & [x, y, z] \\ Relations & : & \left[-y \le -25,\ -y + \frac{13\,z}{5} \le \frac{73}{5},\ y - \frac{z}{2} \le 25,\ x = 0\right] \end{cases}$$

$$point\_in\_DP := \begin{cases} Coordinates & : & [x, y, z] \\ Relations & : & [z = 9,\ y = 29,\ x = 0] \end{cases}$$

$$Region\_in\_sys := \begin{cases} Coordinates & : & [x, y, z] \\ Relations & : & \left[z = 9,\ y = 29,\ -x \le -\frac{37}{2},\ x \le \frac{56}{3}\right] \end{cases}$$



We can see the point (y, z) = (29, 9) is in the real shadow, but not in the dark shadow.
Substitute (y, z) = (29, 9) to the input system, we get 37/2 <= x <= 56/3, which contains no integer value.

2.6 IntegerSolve

This section will introduce the main command of the library, namelyIntegerSolve.The input consists of a system of linear constraints and optional arguments. The system of constrainsts can be given either as two lists (one for linear equations, one for linear inequalities) or two matrices (defining respectively linear equations and linear inequalities). These two different input formats correspond to two different implementations (one using sparse polynomial expressions and one using dense linear algebra) of the same algorithm. The output is a list of Z-polyhedra, the union of which represents the integer points of the input polyhedron. Each of these output Z-polyhedra has at least one integer point and enjoys a property similar to that of a regular chain: every integer point in a projection of that Z-polyhedron can be lifted to an integer point of that Z-polyhedron.

The command takes optional boolean arguments dark_shadow, grey_shadow, HNF and partition, all with default values true. It calls the commands DarkShadow and GreyShadow. If we turn off dark_shadow (set it to be false), this function will only implement the GreyShadow. Same with the grey_shadow. The option HNF is to choose the methods (HNF or Pugh's method) to solve the equations. The options partition is to determine whether we want a disjoint decomposition of the integer points or not.

```
> restart: PLHD:=Polyhedra([x,y,z]): with(PLHD):
  equations := []; inequalities := [3*x-2*y+z<= 7, -2*x+2*y-z
  <= 12, -4*x+y+3*z <= 15, -y <= -25];vars:=[x, y, z];
  res:=IntegerSolve(equations, inequalities, vars, true, true,
  true, true);
  map(Display, res);
```

$$equations := [\,]$$

$$inequalities := [3\,x-2\,y+z\le 7,\ -2\,x+2\,y-z\le 12,\ -4\,x+y+3\,z\le 15,\ -y\le -25]$$

$$vars := [x, y, z]$$

$$res := [Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron]$$

$$\left[\left\{\begin{array}{c} -4\,x+y+3\,z\le 15\\ -2\,x+2\,y-z\le 12\\ 3\,x-2\,y+z\le 7\\ -y\le -25\\ 13\,z-5\,y\le 67\\ 2\,y-z\le 48\\ -z\le -2\\ z\le 17\end{array}\right.,\ \left\{\begin{array}{c} x=15\\ y=27\\ z=16\end{array}\right.,\ \left\{\begin{array}{c} x=18\\ y=33\\ z=18\end{array}\right.,\ \left\{\begin{array}{c} x=14\\ y=25\\ z=15\end{array}\right.,\ \left\{\begin{array}{c} x=19\\ y=50+t\_23\\ z=50+2\,t\_23\\ -t\_23\le 25\\ t\_23\le -16\end{array}\right.\right] \tag{2.6.1}$$

```
> res:=IntegerSolve(equations, inequalities, vars,true, true,
  false, true);
  map(Display, res);
```

$res := [Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron]$

$$\left[\left\{\begin{array}{c} -4x+y+3z \le 15 \\ -2x+2y-z \le 12 \\ 3x-2y+z \le 7 \\ -y \le -25 \\ 13z-5y \le 67 \\ 2y-z \le 48 \\ -z \le -2 \\ z \le 17 \end{array}\right., \left\{\begin{array}{c} x=15 \\ y=27 \\ z=16 \end{array}\right., \left\{\begin{array}{c} x=18 \\ y=33 \\ z=18 \end{array}\right., \left\{\begin{array}{c} x=14 \\ y=25 \\ z=15 \end{array}\right., \left\{\begin{array}{c} x=19 \\ z=-50+2y \\ -y \le -25 \\ y \le 34 \end{array}\right.\right] \quad (2.6.2)$$

```
> res:=IntegerSolve(equations, inequalities,vars, true, true,
  true, false);
  map(Display, res);
```

$res := [Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron]$

(2.6.3)

$$\left[\begin{cases} -4\,x+y+3\,z\le 15 \\ -2\,x+2\,y-z\le 12 \\ 3\,x-2\,y+z\le 7 \\ -y\le -25 \\ 13\,z-5\,y\le 67 \\ 2\,y-z\le 48 \\ -z\le -2 \\ z\le 17 \end{cases}, \begin{cases} x=17 \\ y=31 \\ z=17 \end{cases}, \begin{cases} x=-15+t\_3+t\_4 \\ y=-45+4\,t\_3+t\_4 \\ z=t\_4 \\ -4\,t\_3-t\_4\le -70 \\ -5\,t\_3+2\,t\_4\le -38 \\ 6\,t\_3-t\_4\le 72 \\ -t\_4\le -15 \\ t\_4\le 16 \end{cases}, \right. \tag{2.6.3}$$

$$\begin{cases} x=15 \\ y=27 \\ z=16 \end{cases}, \begin{cases} x=18 \\ y=33 \\ z=18 \end{cases}, \begin{cases} x=14 \\ y=26 \\ z=15 \end{cases}, \begin{cases} x=13 \\ y=25 \\ z=14 \end{cases}, \begin{cases} x=14 \\ y=26 \\ z=15 \end{cases},$$

$$\begin{cases} x=15 \\ y=27 \\ z=16 \end{cases}, \begin{cases} x=1+t\_12 \\ y=-27+4\,t\_12 \\ z=15 \\ -4\,t\_12-t\_13\le -67 \\ -5\,t\_12+2\,t\_13\le -35 \\ 6\,t\_12-t\_13\le 68 \\ -t\_13\le -13 \\ t\_13\le 18 \end{cases}, \begin{cases} x=14 \\ y=25 \\ z=15 \end{cases},$$

$$\left\{\begin{array}{l} x=17 \\ y=31 \\ z=17 \end{array}\right., \left\{\begin{array}{l} x=16 \\ y=30 \\ z=16 \end{array}\right., \left\{\begin{array}{l} x=14 \\ y=25 \\ z=15 \end{array}\right.,$$

$$\left\{\begin{array}{c} x=-12+t\_21 \\ y=-12+2\,t\_21+t\_22 \\ z=-12+2\,t\_21+2\,t\_22 \\ -2\,t\_21-t\_22\leq -37 \\ t\_21\leq 31 \\ t\_21+t\_22\leq 15 \\ 4\,t\_21+7\,t\_22\leq 15 \\ -t\_22\leq 25 \\ t\_22\leq -13 \end{array}\right., \left\{\begin{array}{c} x=-12-t\_23 \\ y=25 \\ z=62+2\,t\_23 \\ -t\_23\leq 31 \\ t\_23\leq -25 \end{array}\right.\Bigg]$$

```
> res:=IntegerSolve(equations, inequalities, vars,true, true,
  false, false);
  map(Display, res);
```

$res := [Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron]$

$$\Bigg[\left\{\begin{array}{c} -4\,x+y+3\,z\leq 15 \\ -2\,x+2\,y-z\leq 12 \\ 3\,x-2\,y+z\leq 7 \\ -y\leq -25 \\ 13\,z-5\,y\leq 67 \\ 2\,y-z\leq 48 \\ -z\leq -2 \\ z\leq 17 \end{array}\right., \left\{\begin{array}{l} x=17 \\ y=31 \\ z=17 \end{array}\right., \left\{\begin{array}{c} y=4\,x-3\,z+15 \\ -4\,x+3\,z\leq -10 \\ -5\,x+7\,z\leq 37 \\ 6\,x-7\,z\leq -18 \\ -z\leq -15 \\ z\leq 16 \end{array}\right., \qquad \textbf{(2.6.4)}$$

$$\left\{\begin{array}{l} x=15 \\ y=27 \\ z=16 \end{array}\right., \left\{\begin{array}{l} x=18 \\ y=33 \\ z=18 \end{array}\right., \left\{\begin{array}{l} x=14 \\ y=26 \\ z=15 \end{array}\right., \left\{\begin{array}{l} x=13 \\ y=25 \\ z=14 \end{array}\right., \left\{\begin{array}{l} x=14 \\ y=26 \\ z=15 \end{array}\right.,$$

$$\left\{\begin{array}{l} x=15 \\ y=27 \\ z=16 \end{array}\right., \left\{\begin{array}{l} x=14 \\ y=25 \\ z=15 \end{array}\right., \left\{\begin{array}{l} x=14 \\ y=25 \\ z=15 \end{array}\right., \left\{\begin{array}{l} x=17 \\ y=31 \\ z=17 \end{array}\right., \left\{\begin{array}{l} x=16 \\ y=30 \\ z=16 \end{array}\right.,$$

$$\left\{\begin{array}{l} x=14 \\ y=25 \\ z=15 \end{array}\right., \left\{\begin{array}{c} z=-2x+2y-12 \\ -10x+7y\leq 51 \\ y-x\leq 15 \\ x\leq 19 \\ -y\leq -25 \\ y\leq 34 \end{array}\right.\Bigg]$$

3. Functions in Matrix form

The commands presented in this section are internal. Nevertheless, they may be of interestt to developers. They support the implementation of IntegerSolve based on dense linear algebra. This latter is likely to be faster than the implementation based on polynomial expression for test-cases where polyhedra have large numbers of facets and where computations tend to densify. In this section, all the three functions have the same input format, that is, ME_in :: {Matrix, list}, MI_in :: {Matrix, list}, where ME_in and MI_in represent systems of equations and inequalities, respectively. See the following example:

```
> ME_in := [];
  MI_in := [[3, -2, 1, 7], [-2, 2, -1, 12], [-4, 1, 3, 15], [0, -1, 0, -25]];
```

$$ME\_in := [\,]$$

$$MI\_in := [[3, -2, 1, 7], [-2, 2, -1, 12], [-4, 1, 3, 15], [0, -1, 0, -25]] \quad \textbf{(3.1)}$$

This is equivalent to input: equations := [ ];
inequalities := [3*x1-2*x2+x3 <= 7, -2*x1+2*x2-x3 <= 12, -4*x1 + x2 +

3*x3 <= 15, -x2 <= -25];
Note that this is the example we used above.

3.1 LAFMelim

Input ME_in :: {Matrix, list}, MI_in :: {Matrix, list}, the function LAFMelim will compute its the real projection over rational numbers, based the Fourier-Motzkin elimination.
Ouptput ME_out :: {Matrix, list} and MI_out :: {Matrix, list}, where ME_out and MI_out represent systems of equations and inequalities, respectively.

```
> PLHD:-LAFMelim(ME_in, MI_in);
```

$$[\ ],\ \begin{bmatrix} 1 & -\frac{2}{3} & \frac{1}{3} & \frac{7}{3} \\ -1 & 1 & -\frac{1}{2} & 6 \\ -1 & \frac{1}{4} & \frac{3}{4} & \frac{15}{4} \\ 0 & 1 & -\frac{1}{2} & 25 \\ 0 & -1 & \frac{13}{5} & \frac{73}{5} \\ 0 & -1 & 0 & -25 \\ 0 & 0 & 1 & \frac{132}{7} \\ 0 & 0 & -1 & 0 \end{bmatrix} \quad (3.1.1)$$

3.2 LACanonicalize

Input ME_in :: {Matrix, list}, MI_in :: {Matrix, list}, the function LAFMelim will solve the equations, remove the redundant inequalites and output a block-triangularlized matrix.
Output ME_out :: {Matrix, list} and MI_out :: {Matrix, list}, where ME_out and MI_out represent systems of equations and inequalities, respectively.

```
> PLHD:-LACanonicalize(ME_in, MI_in);
```

(4.1)

$$[\,],\begin{bmatrix} 3 & -2 & 1 & 7 \\ -2 & 2 & -1 & 12 \\ -4 & 1 & 3 & 15 \\ 0 & 2 & -1 & 50 \\ 0 & -5 & 13 & 73 \\ 0 & -1 & 0 & -25 \\ 0 & 0 & 1 & 18 \\ 0 & 0 & -1 & 0 \end{bmatrix} \tag{4.1}$$

3.3 LAIntegerSolve

Input ME_in :: {Matrix, list}, MI_in :: {Matrix, list}, the function LAIntegerSolve will decompose the polyhedron represented by them into several "simpler" polyhedra, each "simpler" polyhedron contains at least one integer point and all the integer points in the original polyhedron lie in exactly one "simpler" polyhedron.

```
> dec := PLHD:-LAIntegerSolve(ME_in, MI_in);
  map(PLHD:-Display, dec);
```

$$dec := [Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron, Z\_polyhedron]$$

$$\left[\begin{cases} 3x-2y+z \le 7 \\ -2x+2y-z \le 12 \\ -4x+y+3z \le 15 \\ 2y-z \le 48 \\ 13z-5y \le 67 \\ -y \le -25 \\ z \le 17 \\ -z \le -2 \end{cases}, \begin{cases} x = 19 \\ y = t\_1 + 50 \\ z = 2\,t\_1 + 50 \\ t\_1 \le -16 \\ -t\_1 \le 25 \end{cases}, \begin{cases} x = 15 \\ y = 27 \\ z = 16 \end{cases}, \begin{cases} x = 18 \\ y = 33 \\ z = 18 \end{cases}, \begin{cases} x = 14 \\ y = 25 \\ z = 15 \end{cases} \right] \tag{5.1}$$

```
> time(PLHD:-LAIntegerSolve(ME_in, MI_in));
```

$$0.826 \tag{5.2}$$

```
> equations := []; inequalities := [3*x-2*y+z<= 7, -2*x+2*y-z <=
  12, -4*x+y+3*z <= 15, -y <= -25];vars:=[x, y, z];
  time(PLHD:-IntegerSolve(equations, inequalities));
```

$$equations := [\ ]$$

$$inequalities := [3\,x-2\,y+z \leq 7,\ -2\,x+2\,y-z \leq 12,\ -4\,x+y+3\,z \leq 15,\ -y \leq -25]$$

$$vars := [x, y, z]$$

$$2.442 \tag{5.3}$$