Chapter 1
Introduction

1.1 Definition of Lattice and Examples

A *lattice* is a *discrete* Abelian group, that is, is a non empty set G in which a binary operation $+$ is defined, with the following properties:

- $u+v=v+u$, for all $u,v \in G$,
- $u+(v+z)=(u+v)+z$, for all $u,v,z \in G$,
- G contains an *identity* element, denoted by 0, such that $u+0=u, u \in G$,
- to each $u \in G$ there corresponds an element $-u \in G$, such that $u-u=0$, where $u-u$ stands for $u+(-u)$.

Discrete means that the set G is enumerable.

The most important classes of lattices is given by subgroups of $\mathbb{R}$ and in general by subgroups of $\mathbb{R}^m$, where the binary operation is the ordinary addition and in general its $m$dimensional extension.

The simplest example of lattice is the set of integer $\mathbb{Z}$ or its "dimensional" version $\mathbb{Z}(d) \stackrel{\Delta}{=} \{nd|n \in \mathbb{Z}\}$, where d is a positive real number. These are 1D lattices. In the mD case the simplest example is $\mathbb{Z}^m$ or its "dimensional" version $\mathbb{Z}(d_1,\ldots,d_m) \stackrel{\Delta}{=} \{(n_1d_1,\ldots,n_md_m)|(n_1,\ldots,n_m) \in \mathbb{Z}^m\}$, where the d_i are positive real numbers. This lattice, illustrated in Fig.1.1 for $m=2$ and $m=3$, is *separable*.

In general, an mD lattice is generated in the form (see *Unified Signal Theory*, Section 3.3)

$$G = \mathbf{G}\mathbb{Z}^m \Longleftrightarrow G = \{\mathbf{G}\,\mathbf{h} \mid \mathbf{h} \in \mathbb{Z}^m\} \tag{1.1}$$

where $\mathbf{G}$ is a non–singular real matrix of dimension $m \times m$. The matrix $\mathbf{G}$ is called a *basis* of the lattice G.

The generation according to (1.1) is typical for *nonseparable* lattices. The simplest example of nonseparable lattice is the *quincunx* lattice, denoted by $\mathbb{Z}_2^1(d_1,d_2)$. This 2D lattice is illustrated in Fig.1.2 and is generated from the basis

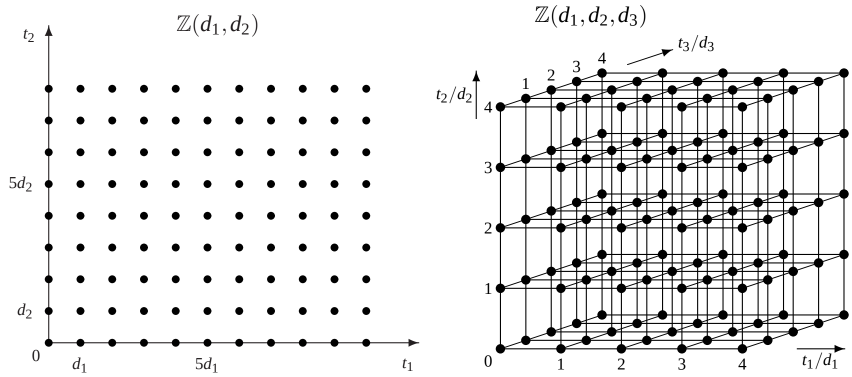
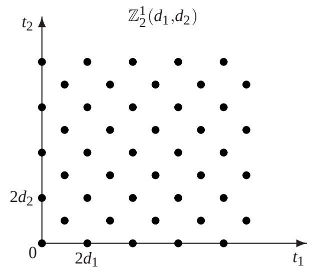


Fig. 1.1 The separable lattices $\mathbb{Z}(d_1,d_2)$ and $\mathbb{Z}(d_1,d_2,d_3)$

Fig. 1.2 The quincunx lattice $\mathbb{Z}_2^1(d_1,d_2)$



$$\mathbf{G}_2 = \begin{bmatrix} d_1 & 0 \\ d_2 & 2d_2 \end{bmatrix} . \tag{1.2}$$

Other nonseparable 2D lattices, which are sublattices of $\mathbb{Z}(d_1,d_2)$, are denoted by $\mathbb{Z}_i^b(d_1,d_2)$, are generated by the bases

$$\mathbf{G} = \begin{bmatrix} id_1 & 0 \\ bd_1 & d_2 \end{bmatrix} = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix} \begin{bmatrix} i & 0 \\ b & 1 \end{bmatrix} \tag{1.3}$$

where i and b are natural numbers with $b < i$. For the illustration of these lattices see the gallery of Fig. 3.11 of the *Unified Signal Theory*.

Fig.1.3 shows two examples of 3D nonseparable lattices, which are both sublattices of $\mathbb{Z}(d_1, d_2, d_3)$.

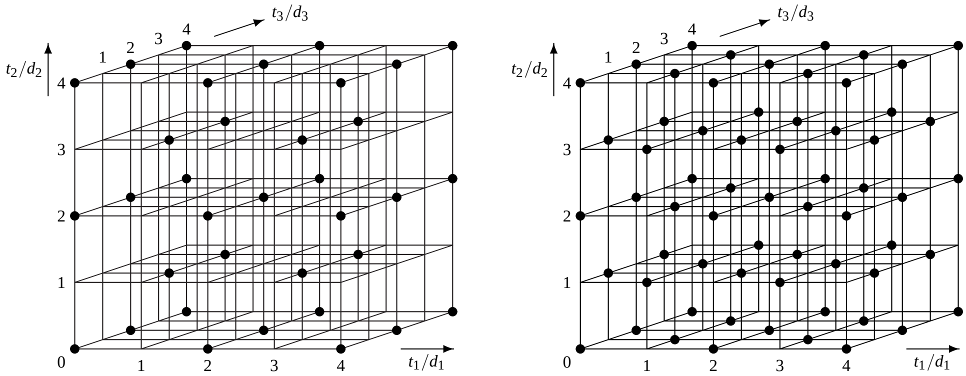


Fig. 1.3 Examples of sublattices of $\mathbb{Z}(d_1, d_2, d_3)$

The lattices generated in the form are called *full-dimensional*. The *reduced-dimensional* lattices can be generated in the form

$$G = \mathbf{G}\,\mathbb{Z}^r \times \mathbb{O}^s \Longleftrightarrow G = \{\mathbf{G}\,\mathbf{h} \mid \mathbf{h} \in \mathbb{Z}^r \times \mathbb{O}^s\} \tag{1.4}$$

where $\mathbb{O} = \{0\}$ is the trivial 1D group, and r and s are natural numbers with $r + s = m$. The dimensionality of these lattices is r, but they are displayed in an mD space and, for this reason, they are called rD lattice in $\mathbb{R}^m$. Examples of 1D lattices in $\mathbb{R}^2$ are shown in Fig.1.4.

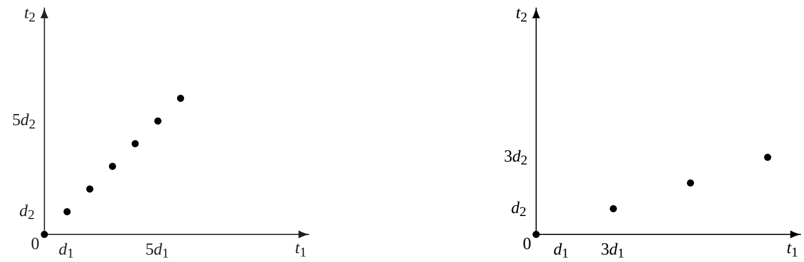


Fig. 1.4 Examples of lattices 1D in 2D

An alternative form of generating reduced-dimensional lattices, starting from a matrix of rank r, will be seen in Chap.3.

1.2 Importance of Lattices

Lattices find applications in several fields. Here we limit ourselves to consider the applications they have in Signal Theory.

Lattices represents the *domains* of discrete signals and the *periodicity* of periodic signals, as discussed in great detail in Chapter 5 of the *Unified Signal Theory*. The domains are always full-dimensional groups, and in particular full-dimensional lattices for discrete signals. The periodicity can be represented by a full-dimensional lattice (full periodicity) but also by a reduced-dimensional lattice (partial periodicity).

Lattices play also a fundamental role in the formulation of *cells*, both aperiodic and periodic cells (see Chapter 3 of the *Unified Signal Theory*).

Lattices are used in the *reciprocal* form in the frequency domain, where they represent the domain and the periodicity of the Fourier transform, exactly as for the signals (see Chapter 3 of the *Unified Signal Theory*).

Finally, we recall the role of lattices in linear transformations and in particular in *multirate systems* (see Chapters 6 and 7 of the *Unified Signal Theory*).

1.3 Why a Computer Program for Lattices

The representation of a lattice through a basis, according to (1.1) and (1.4), is not unique and two distinct bases of a same lattice are related by an integer matrix with unitary determinant, called *unimodular* matrix. From Section 3.3 of the *Unified Signal Theory*, we recall the following statements:

Proposition 1.1. *If G is a lattice with basis $\mathbf{G}$, all the other bases have the form $\mathbf{GE}$, where $\mathbf{E}$ is any matrix of integers $\mathbf{E}$ such that $\det \mathbf{E} = \pm 1$ (unimodular matrix).*

Proposition 1.2. *The bases of the sublattices G of a given lattice G_0 can be generated in the form*

$$\mathbf{G} = \mathbf{G}_0 \mathbf{A}\,, \tag{1.5}$$

where $\mathbf{G}_0$ is a basis of G_0, and $\mathbf{A}$ is a non–singular matrix of integers. If $|\det \mathbf{A}| = 1$, then $G = G_0$, if $|\det \mathbf{A}| > 1$, G is a proper sublattice of G_0 (note that $\det \mathbf{A}$ is always an integer).

Now it is evident the role of integer matrices in lattice representation. It is also evident, from the multiplicity of the bases, the convenience of searching for efficient bases to economize lattice representation. As a matter of fact, it is possible to get simple bases using the procedures of triangularization (due to Hermite) and of diagonalization (due to Smith). These procedures, which are fundamental topics in the theory of integer matrices, require a manipulation of an integer matrix, which is based on *elementary operations* on the columns and sometimes on the rows of the given integer matrices. Of course, this can be done by hand but becomes soon stressful and the help of the computer is mandatory.

Other important operations on lattices are the *sum* and *intersection* (see Chapter 16 of the *Unified Signal Theory*). Also for these operations the manipulation of integer matrices are needed and a computer program is the appropriate solution.

1.4 Procedures Implemented with `Mathematica`

The procedures implemented with `Mathematica` are concerned with the following topics related to lattices:

- triangularization and diagonalization of integer matrices;
- evaluation and classification of the points of an mD lattices in a gives *region* of the mD space $\mathbb{R}^m$;
- sum and intersection of lattices;
- alignment of the bases of a lattice and of sublattice;
- continuous cells;
- discrete cells.

The corresponding procedures (macros) are illustrated in separated chapters of this book.

The `Mathematica` code of the procedures is in the file `lattices.m`

1.4.1 Demos with combination of `Mathematica` *and TEX*

Each group of the procedures is illustrated with specific demos, which are contained in the files:

- `demo1macro.m`: demos of triangular and diagonal decomposition;
- `demo2macro.m`: demos of lattice points;
- `demo3macro.m`: macros for demos of sum and intersection of two lattices;
- `demo4macro.m`: demos of base alignment of a lattice and a sublattice;
- `demo5macro.m`: demos of continuous cells;
- `demo6macro.m`: demos of discrete cells.

The demos are obtained with combination of `Mathematica` and TEX and in most of the cases produce also a graphic illustration of the results.

To get the demos so many files are available `demo`$\langle i \rangle$`.m`, where the specific data must be introduced. For instance, to get a demo of `demo3macro.m` the file `demo3.m` must be used.

In general, the convention is to use `write`$\langle$`name`$\rangle$ to get the demo $\langle$`name`$\rangle$ and to use `show`$\langle$`name`$\rangle$ to get also the graphic illustration.

1.5 Organization of the Software

The user may follow two different lines

1.5.1 *Use of only* `Mathematica`

In this case the user applies the procedures contained in the file `lattices.m` and described in the following chapters. The results are obtained in the format of `Mathematica` and can be handled for the user's specific application.

It must be noted that the procedures was written in `Mathematica version 3.0 for Linux` and the compatibility with subsequent versions must be checked.

1.5.2 *Combination of* `Mathematica` *and TEX for demos*

In this case the organization may be the following.

Initialization

Copy the package `mathtex` from the website into your home directory `mathtex`.

Then, the directory `mathtex` will contain the following files and directories

- `Mathematica` files

```
lattices.m
demo1macros.m    demo1.m    demo1.fig    directory EL
demo2macros.m    demo2.m    demo2.fig    directory LP
demo3macros.m    demo3.m    demo3.fig    directory SI
demo4macros.m    demo4.m    demo4.fig    directory AL
demo5macros.m    demo5.m    demo5.fig    directory CC
demo6macros.m    demo6.m    demo6.fig    directory CD
```

- TEX files

```
inma2mt.sty  (authors' personal macros)
afig1mt.st   (authors' personal macros)
retic2.tex   (authors' personal macros)
domt.tex
ret.tex
```

In particular the file `domt.tex` contains the specification of the LATEX documentclasses, namely

```
\documentclass[11pt]{book}
\usepackage{latexsym,epic,eepic,psfig, epsfig,graphics}
\usepackage{inma2mt}
\usepackage{afig1mt}
```

The file `domt.tex` is called by the command `calltex` (contained in `latttices.m`), which consistsof the commands

```
calltex:=Module[{st},
  st="! latex domt  \n\n";
  Run[st];
  st="! dvips domt.dvi  \n\n";
  Run[st];
  st="! ghostview domt.ps \n\n";
  Run[st]
]
```

Specific application

Choose an application, for instance “triangular and diagonal decomposition”. Then, write in `demo1.m` the specific demo, which may be

```
lab="EL3"
initdemo1[lab]
a33={{1,2,3},
   {3,0,2},
   {4,-1,3}}
showtriangall[a33,1]
completedemo1
```

Then, writing in `Mathematica` the command

```
<<demo1.m
```

the user will see in the preview the specific demo and in the directory `EL` the file `EL3.tex` with the TEX code and the postscript file `EL3.ps`.

1.6 List of the Main Procedures

1.6.1 Triangularization and Diagonalization (Chapter 2)

Table 1.1 Triangularization and Diagonalization

Procedure	Purpose

`leftUfactor[a,adj]`
`leftLfactor[a,adj`
`rightUfactor[a,adj`
`rightLfactor[a,adj`
	given a basis `a` (integer matrix), gives a unimodular matrix `e`, such that `a.e` or `e.a` gives the triangular form

`hermite[a,form,adj]`	decomposes the matrix $a = \mathbf{A}$ in the product of two matrices $\mathbf{A} = \mathbf{E}\mathbf{U}$ if form=1, $\mathbf{A} = \mathbf{E}\mathbf{L}$ if form=2, $\mathbf{A} = \mathbf{U}\mathbf{E}$ if form=3, $\mathbf{A} = \mathbf{U}\mathbf{E}$ if form=4, where $\mathbf{U}$ is upper triangular and $\mathbf{L}$ is lower triangular.
`triangU[a,adj]` `triangL[a,adj]`	evaluate the triangular forms of the matrix `a`
`smith[a,adj]`	decomposes the $a = \mathbf{A}$ in the form $\mathbf{A} = \mathbf{E}_1 \blacksquare \mathbf{E}_2$, where $\blacksquare$ is diagonal, $\mathbf{E}_1$, $\mathbf{E}_2$ are unimodular

1.6.2 Generation of Lattice Points (Chapter 3)

Table 1.2 Generation of Lattice Points

Procedure	Purpose
`latticepoints2D[a,region,op]`	evaluates the points of the 2D lattice specified by the matrix `a` belonging to the set of $\mathbb{R}^2$ `region`
`latticepoints3D[a,region,op]`	evaluates the points of the 3D lattice specified by the matrix `a` belonging to the set of $\mathbb{R}^3$ `region`
`reglatticepoints[a,region,op]`	evaluates the points of the mD lattice specified by the matrix `a` belonging to the set of $\mathbb{R}^m$ `region`; the lattice is assumede regular (full-dimensional)
`deglatticepoints[a,region,op]`	evaluates the points of a degenerate mD lattice (rduced-dimensional) specified by the matrix `a` belonging to the set of $\mathbb{R}^m$ `region`
`latticepoints[a,region,op]`	unifies all the above procedures (the lattice may be full-dimensional, as well reduced dimensional

1.6.3 Sum and Intersection of Two Lattices (Chapter 4)

Table 1.3 Sum and Intersection of Two Lattices

Procedure	Purpose
`lcrmGCLD[a,b,adj]`	given the bases `a` and `b` of two lattices, evaluates the bases of the sum and of the intersection

1.6.4 Basis Alignment (Chapter 5)

Table 1.4 Alignmment of the bases of a lattice and a sublattice

Procedure	Purpose
`basealignment[a,b]`	given the bases of a lattice and a sublattice, evaluates the aligned bases

1.6.5 Continuous Cells (Chapter 6)

Table 1.5 Continuous Cells

Procedure	Purpose
`cell2D[a,type]`	gives the vertexes of a 2D cell $\mathbb{R}^2/L$, where L is a lattice specified by the basis `a`; 9 types of 2D cells are available
`cell3D[a,type,cent,vp]`	gives the faces of a 3D cell $\mathbb{R}^2/L$, where L is a lattice specified by the basis `a`; four types of 3D cells ara available
`voronoi3D[a]`	gives the faces of a 3D Voronoi cell (tt a specifies e 3D lattice)

1.6.6 Discrete Cells (Chapter 7)

Table 1.6 Discrete Cells

Procedure	Purpose
`discretecell2D[a,b,type]`	gives the points of a 2D cell L_a/L_b, where L_a is a lattice specified by the basis `a` and L_b is a sublattice specified by the basis `b` 9 types of 2D cells are available
`cell3D[a,type,cent,vp]`	gives the points of a 3D cell L_a/L_b, where L_a is a lattice specified by the basis `a` and L_b is a sublattice specified by the basis `b`; 4 types of 2D cells are available
`discretecellvoronoi3D[a,b]`	gives the points of a 3D Voronoi cell; (`a` specifies a 3D lattice), `b` specifies a 3D sublattice)

Chapter 2
Triangularization and Diagonalization of Integer Matrices

2.1 Introduction

The topics of this chapter are developed in Chapter 16 of the *Unified Signal Theory*, in particular the *triangularization* according to the Hermite decomposition and the *diagonalization* according to the Smith decomposition. These decompositions are obtained by *elementary operations* on the columns and sometimes on the rows of the given integer matrix.

The fundamental procedures implemented with `Mathematica` are:

`hermite[a]`,
`tringU[a]` and `tringL[a]`,
`smith[a]`,

where `a` is the integer matrix. They are presented with the following schedule:

- usage with `Mathematica`,
- demo with `Mathematica` and TEX,

whereas the description of the `Mathematica` program is developed in the final section of the chapter. Thus, the reader can use and gets confidence with the fundamental procedures without entering into the detail of the program.

Normally the above procedures are applied to square matrices, as happens for lattices, but their application to rectangular matrices is also possible. This will be discussed in a separate section.

The complete list of the procedures developed in this chapter are:

- `hermite[a]` (Sect. 2.3),
- `tringU[a]` and `tringL[a]` (Sect. 2.3),
- `tringPU[a]` and `tringPL[a]` (Sect. 2.4),
- `smith[a]` (Sect. 2.5),
- `annihilate`: is the fundamental macro that operates on two rows of the given matrix; its iterative application allows the implementation of the other macros (Sect. 2.8).

- `leftUfactor[a]`: provides the upper–triangular form of the given matrix (Sect. 2.8).
- `setcanonic[a]`: provides the canonical form starting from the upper–triangular form (Sect. 2.8).
- `leftLfactor[a]`
- `rightLfactor[a]`
- `rightUfactor[a]`

All these procedures are contained in the file `lattices.m`.

2.2 Definitions and Operations on Integer Matrices

From Section 16.6 of the *Unified Signal Theory* we recall the following symbols:

- $\mathcal{I}_{mn}$ denotes the class of integer matrices of dimensions $m \times n$,
- $\mathcal{I}_m$ denotes the class of integer square matrices of dimensions $m \times m$
- $\mathcal{U}_m$ denotes the class of *unimodular* integer matrices (a matrix $\mathbf{E} \in \mathcal{I}_m$ is unimodular if $\det \mathbf{E} = \pm 1$).

2.2.1 Triangular and diagonal matrices. Canonical forms

A matrix $\mathbf{A} = \|a_{ij}\|_{m \times n} \in \mathcal{I}_{mn}$ is:

- *upper–triangular* (type **U**) if $a_{ij} = 0$ for $i > j$,
- *lower–triangular* (type **L**) if $a_{ij} = 0$ for $i < j$,
- *diagonal* (type $\boldsymbol{\Delta}$) if $a_{ij} = 0$ $i \neq j$.

Examples are

$$\begin{bmatrix} 3 & 2 & 1 \\ 0 & 2 & -1 \end{bmatrix} \qquad \begin{bmatrix} 3 & -1 \\ 0 & 2 \\ 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 3 & -1 & 2 \\ 0 & 2 & 1 \\ 0 & 0 & 3 \end{bmatrix} \qquad \text{type } \mathbf{U}$$

$$\begin{bmatrix} 3 & 0 & 0 \\ -1 & 2 & 0 \end{bmatrix} \qquad \begin{bmatrix} 3 & 0 \\ 4 & 2 \\ 1 & -1 \end{bmatrix} \qquad \begin{bmatrix} 3 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 1 & 2 \end{bmatrix} \qquad \text{type } \mathbf{L}$$

$$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix} \qquad \begin{bmatrix} 3 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{type } \boldsymbol{\Delta}$$

Normally these definitions are applied to square matrices, but sometimes also to rectangular matrices.

For square matrices the upper–triangular and lower-triangular forms have respectively the explicit structures

$$\mathbf{U} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ 0 & u_{22} & \cdots & u_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{mm} \end{bmatrix} \qquad \mathbf{L} = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{m1} & l_{m2} & \cdots & l_{mm} \end{bmatrix} \tag{2.1}$$

and we can choose for **U** and **L** the **canonical** form, having the following constraints (when the matrices are nonsingular):

$$\begin{aligned} &u_{ii} > 0\,, \qquad 0 \le u_{ij} < u_{ii} \quad (j > i) \quad \text{with } u_{ii} \text{ and } u_{ij} \text{ coprime} \\ &l_{ii} > 0\,, \qquad 0 \le l_{ij} < l_{ii} \quad (j < i) \quad \text{with } l_{ii} \text{ and } l_{ij} \text{ coprime.} \end{aligned} \tag{2.2}$$

2.2.2 Elementary Operations on Integer Matrices

Elementary operations on the class $\mathcal{I}_{mn}$ provide the rearrangement of an integer matrix to obtain triangular and diagonal forms. Given an $m \times n$ integer matrix **A**, the *elementary operations on the columns* are:

1) permutation of two columns,
2) multiplication of a column by -1,
3) replacement of a column by the sum of itself and an integer k multiple of any other column.

For instance, if $\mathbf{A} = [\mathbf{a}_1\, \mathbf{a}_2\, \mathbf{a}_3]$ is an $m \times 3$ matrix with columns $\mathbf{a}_1$, $\mathbf{a}_2$ and $\mathbf{a}_3$, examples of 1), 2) and 3) are respectively

$$[\mathbf{a}_3\, \mathbf{a}_2\, \mathbf{a}_1]\,, \qquad [\mathbf{a}_1\, -\mathbf{a}_2\, \mathbf{a}_3]\,, \qquad [\mathbf{a}_1\, \mathbf{a}_2\, \mathbf{a}_3 + k\,\mathbf{a}_1]\ .$$

We can check that the modified matrices can be obtained from **A** by a *right multiplication by a unimodular matrix*. In fact

$$\begin{aligned} [\mathbf{a}_3\, \mathbf{a}_2\, \mathbf{a}_1] &= [\mathbf{a}_1\, \mathbf{a}_2\, \mathbf{a}_3] \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \\ [\mathbf{a}_1\, -\mathbf{a}_2\, \mathbf{a}_3] &= [\mathbf{a}_1\, \mathbf{a}_2\, \mathbf{a}_3] \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ [\mathbf{a}_1\, \mathbf{a}_2\, \mathbf{a}_3 + m\,\mathbf{a}_1] &= [\mathbf{a}_1\, \mathbf{a}_2\, \mathbf{a}_3] \begin{bmatrix} 1 & 0 & k \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

In a similar way we introduce the *elementary operations on the rows* of a matrix **A**, which correspond to a *left multiplication by a unimodular matrix*.

As a fundamental application of elementary operations we recall Proposition 16.3 of the *Unified Signal Theory*:

Proposition 2.1. *Let* **G** *be a basis of a lattice G. Then, elementary operations on the columns of* **G** *provides new bases of G.*

2.3 Hermite Triangularization

From Section 16.6 of the *Unified Signal Theory* we recall:

Theorem 2.1. *An integer matrix* $\mathbf{H} \in \mathfrak{I}_{mn}$ *can be decomposed in the form*

$$\underset{m\times n}{\mathbf{H}} = \underset{m\times n}{\mathbf{U}}\ \underset{n\times n}{\mathbf{E}_1} = \underset{m\times n}{\mathbf{L}}\ \underset{n\times n}{\mathbf{E}_2}$$

where **U** *is upper–triangular,* **L** *is lower–triangular and* $\mathbf{E}_1, \mathbf{E}_2$ *are unimodular.*

These decompositions are obtained by applying elementary operations on the columns of the given integer matrix **H**.[1]

2.3.1 Triangularization by hand

To see how the triangularization is time consuming, we develop an example by hand, considering a square matrix, which is the case of main interest.

Example 2.1. Consider the 3×3 matrix

$$\mathbf{H} = \begin{bmatrix} 3 & -1 & 4 \\ 2 & 1 & -3 \\ 0 & 2 & -1 \end{bmatrix}.$$

Then

- summing to the second column the third multiplied by 2,
- summing to the second column the first multiplied by 3,
- summing to the first column the second multiplied by -2,
- summing to the third column the second multiplied by 3,
- changing the signs in the first and in the third columns
- summing to the third column to the first multiplied by 2,

one gets in the order

[1] Elementary operations on the rows are also possible, leading to a different kind of triangularization (see Sect. 2.6).

$$\mathbf{H} \rightarrow \begin{bmatrix} 3 & 7 & 4 \\ 2 & -5 & -3 \\ 0 & 0 & -1 \end{bmatrix} \rightarrow \begin{bmatrix} 3 & 16 & 4 \\ 2 & 1 & -3 \\ 0 & 0 & -1 \end{bmatrix} \rightarrow \begin{bmatrix} -29 & 16 & 4 \\ 0 & 1 & -3 \\ 0 & 0 & -1 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} -29 & 16 & 52 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \rightarrow \begin{bmatrix} 29 & 16 & -52 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 29 & 16 & 6 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{U}$$

where $\mathbf{U}$ has the upper–triangular canonical form, that is with constraints (2.2). In the same way one gets the lower–triangular form, which results

$$\mathbf{H} \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 5 & 7 & 29 \end{bmatrix} = \mathbf{L} \quad .$$

2.3.2 The Mathematica procedure "hermite"

Usage

```
{unimod,triang} = hermite[a,form,adj]
```

where

- `a` is the given integer matrix,
- `form` specifies the form of triangularization, according to
  `form` = 1 $\mathbf{A} = \mathbf{U}\mathbf{E}$
  `form` = 2 $\mathbf{A} = \mathbf{L}\mathbf{E}$
  `form` = 3 $\mathbf{A} = \mathbf{E}\mathbf{U}$
  `form` = 4 $\mathbf{A} = \mathbf{E}\mathbf{L}$

where $\mathbf{U}$ is upper triangular, $\mathbf{L}$ is lower triangular, $\mathbf{E}$ is unimodular,

- `adj` is an it optional argument with default value `adj = 1`, which provides the canonical form, while with `adj = 0` $\mathbf{U}$ and $\mathbf{L}$ are not necessarily in canonical form,
- `unimod` is the unimodular matrix,.
- `triang` is the triangular matrix $\mathbf{U}$ or $\mathbf{L}$.

Hence, one gets the eight possible versions

```
{r,s} = hermite[a,1]      canonical
{r,s} = hermite[a,2]      canonical
{r,s} = hermite[a,3]      canonical
{r,s} = hermite[a,4]      canonical
{r,s} = hermite[a,1,0]
{r,s} = hermite[a,2,0]
```

```
{r,s} = hermite[a,3,0]
{r,s} = hermite[a,4,0]
```

The forms 3 and 4, operating on the rows instead of columns, will be discussed later.

2.3.3 *The Mathematica procedures "triangU" and "triangL"*

Usage

```
b =  triangU[a,adj]
b =  triangL[a,adj]
```

where

- `a` is the given integer matrix,
- `adj` is an *optional argument* with default value `adj = 1`, which provides the canonical form, while with `adj = 0` the form is not necessarily canonical,
- `b` is the triangular form.

Therefore, one has the possible forms:

```
b = triangU[a]      canonical
b = triangU[a,0]
b = triangL[a]      canonical
b = triangL[a,0]
```

Remark. The statement `b=triangU[a]` is equivalent to the statements

```
{unimod,triang}=hermite[a,1]
b=triang
```

and similar considerations hold for `triangL`.

2.3.4 Demo of matrix triangularizations

Given a matrix $\mathbf{A}$ of dimensions $m \times n$, the demo procedure $\boxed{\texttt{showtriangall}}$ evaluates and writes the matrices $\mathbf{E}, \mathbf{U}$ and $\mathbf{L}$ such that the following decompositions hold:

$$\underset{m\times n}{\mathbf{A}}\ \underset{n\times n}{\mathbf{E}} = \underset{m\times n}{\mathbf{U}} \qquad \mathbf{E} \text{ right unimodular factor} \quad \mathbf{U} \text{ upper triangular}$$

$$\underset{m\times n}{\mathbf{A}}\ \underset{n\times n}{\mathbf{E}} = \underset{m\times n}{\mathbf{L}} \qquad \mathbf{E} \text{ right unimodular factor} \quad \mathbf{L} \text{ lower triangular}$$

$$\underset{m\times m}{\mathbf{E}}\ \underset{m\times n}{\mathbf{A}} = \underset{m\times n}{\mathbf{U}} \qquad \mathbf{E} \text{ left unimodular factor} \quad \mathbf{U} \text{ upper triangular}$$

$$\underset{m\times m}{\mathbf{E}}\ \underset{m\times n}{\mathbf{A}} = \underset{m\times n}{\mathbf{L}} \qquad \mathbf{E} \text{ left unimodular factor} \quad \mathbf{L} \text{ lower triangular}$$

To get the demo the following is needed in `Mathematica`

- 1) define the integer matrix `a`
- 2) write the statement `showtriang[a]`.

Example 2.2. The statements for `Mathematica` are

```
Get["demo1macro.m"]
lab="EL101"; initdemo1[lab]
a44={{1,2,3,5},
    {3,0,2,-1},
    {4,-1,3,-7},
     {1,3,5,7 }}
writehermiteall[a44,0]
completedemo1
```

Hence `Mathematica` and TEX give

Triangularization: run $\boxed{\textbf{EL101}}$ **(May 26, 2011)**
Given an integer matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & 5 \\ 3 & 0 & 2 & -1 \\ 4 & -1 & 3 & -7 \\ 1 & 3 & 5 & 7 \end{bmatrix}$$

the procedure `writehermiteall` writes the decompositions

$$\mathbf{A\,E} = \mathbf{L} \qquad \mathbf{A\,E} = \mathbf{U} \qquad \mathbf{E\,A} = \mathbf{L} \qquad \mathbf{E\,A} = \mathbf{U}$$

where **E** are unimodular, **L** are lower triangular **U** are upper triangular.
If `adj = 1` the triangular form is "adjusted" to have a canonical form.

---

matrix 4×4. `adj=0` $\mathbf{A\,E} = \mathbf{L}$

$$\begin{bmatrix} 1 & 2 & 3 & 5 \\ 3 & 0 & 2 & -1 \\ 4 & -1 & 3 & -7 \\ 1 & 3 & 5 & 7 \end{bmatrix} \begin{bmatrix} 1 & -1 & 4 & -1 \\ 0 & -1 & 7 & -5 \\ 0 & 1 & -6 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 3 & -1 & 0 & 0 \\ 4 & 0 & -9 & 0 \\ 1 & 1 & -5 & 1 \end{bmatrix}$$

matrix 4×4. `adj=0` $\mathbf{A\,E} = \mathbf{U}$

$$\begin{bmatrix} 1 & 2 & 3 & 5 \\ 3 & 0 & 2 & -1 \\ 4 & -1 & 3 & -7 \\ 1 & 3 & 5 & 7 \end{bmatrix} \begin{bmatrix} -14 & 0 & 3 & 1 \\ -81 & 4 & 4 & 0 \\ 29 & -1 & -3 & 0 \\ 16 & -1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -9 & 0 & 2 & 1 \\ 0 & -1 & 3 & 3 \\ 0 & 0 & -1 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

matrix 4×4. `adj=0` $\mathbf{E\,A} = \mathbf{L}$

$$\begin{bmatrix} 14 & 0 & 1 & -9 \\ -5 & -4 & 1 & 4 \\ 5 & -17 & 6 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 & 5 \\ 3 & 0 & 2 & -1 \\ 4 & -1 & 3 & -7 \\ 1 & 3 & 5 & 7 \end{bmatrix} = \begin{bmatrix} 9 & 0 & 0 & 0 \\ -9 & 1 & 0 & 0 \\ -22 & 4 & -1 & 0 \\ 3 & 0 & 2 & -1 \end{bmatrix}$$

matrix 4×4. `adj=0` $\mathbf{E\,A} = \mathbf{U}$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 \\ -5 & 2 & -1 & 3 \\ -16 & 9 & -5 & 9 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 & 5 \\ 3 & 0 & 2 & -1 \\ 4 & -1 & 3 & -7 \\ 1 & 3 & 5 & 7 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 5 \\ 0 & 1 & 2 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 9 \end{bmatrix}$$

In this case `adj=0` and the forms are not canonical.

Given a matrix **A** of dimensions $m \times n$, the demo procedure `writetriangall` evaluates and write the upper and lower triangular forms of **A** obtained with the Hermite decomposition.

To get the demo the following is needed in `Mathematica`:

1) define the integer matrix `a`
2) write the statement `writetriangall[a]`.

Example 2.3. This example of demo consider a 3×3 matrix. The statements for `Mathematica` are

```
Get["demo1macro.m"]
lab="EL102"
initdemo1[lab]
a33={{1,2,3},
    {3,0,2},
    {4,-1,3}}
writetriangall[a33,1]
completedemo1
```

Hence `Mathematica` and TEX give

Triangularization: run EL102 (May 26, 2011)
Given an integer matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 0 & 2 \\ 4 & -1 & 3 \end{bmatrix}$$

the procedure `writetriangall` writes the triangular forms

$$\mathbf{U} = \mathbf{A}\,\mathbf{E}_u \qquad = \mathbf{L} = \mathbf{A}\mathbf{E}_l$$

obtained with the Hermite decomposition (**U** is upper triangular and **L** is lower triangular)
If `adj = 1` the triangular form is “adjusted” to have a canonical form

---

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 0 & 2 \\ 4 & -1 & 3 \end{bmatrix} \rightarrow \mathbf{U} = \begin{bmatrix} 9 & 0 & 7 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 0 & 2 \\ 4 & -1 & 3 \end{bmatrix} \rightarrow \mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 4 & 0 & 9 \end{bmatrix}$$

In this case `adj=1` and therefore the triangular forms are canonical.

2.3.5 Application to lattices

The above triangularizations can be applied directly to the class of "adimensional lattices" $\mathcal{L}_m(\mathbb{Z}^m)$ with an arbitrary dimension m, whose basis is an integer matrix.

For the class of "dimensional lattices", we recall from Section 16.5 of the *Unified Signal Theory*, that for the lattices of a generic class $\mathcal{L}_m(G_0)$ generated by an m–dimensional lattice G_0, it is always possible to write their bases in the form

$$\mathbf{J} = \mathbf{G}_0\,\mathbf{H}$$

with $\mathbf{G}_0$ a basis of G_0 and $\mathbf{H} \in \mathcal{I}_m$. Then, the triangularization can be applied to the integer matrix $\mathbf{H}$ to get the upper and the lower triangular forms, that is,

$$\mathbf{U} = \texttt{triangU}[\mathbf{H}]\,, \qquad \mathbf{L} = \texttt{triangL}[\mathbf{H}]\,.$$

Then the triangularization is transferred to lattices basis as

$$\mathbf{J}_u = \mathbf{G}_0\,\mathbf{U}\,, \qquad \mathbf{J}_l = \mathbf{G}_0\,\mathbf{L}\,. \tag{2.3}$$

Example 2.4. Consider the 3D lattice J with basis

$$\mathbf{J} = \begin{bmatrix} 6d_1 & 2d_1 & 2d_1 \\ 0 & 4d_2 & 2d_2 \\ 6d_3 & 0 & 9d_3 \end{bmatrix}$$

where d_i are arbitrary positive real numbers. This lattice is a sublattice of the separable lattice $G_0 = \mathbb{Z}(d_1, d_2, d_3)$ with basis

$$\mathbf{G}_0 = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix}\,.$$

Then, the basis (2.3) can be written in the form

$$\mathbf{J} = \begin{bmatrix} 6d_1 & 2d_1 & 2d_1 \\ 0 & 4d_2 & 2d_2 \\ 6d_3 & 0 & 9d_3 \end{bmatrix} = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} \begin{bmatrix} 6 & 2 & 2 \\ 0 & 4 & 2 \\ 6 & 0 & 9 \end{bmatrix}$$

and the application of `triangularU` to the latter matrix gives

$$\mathbf{H} = \begin{bmatrix} 6 & 2 & 2 \\ 0 & 4 & 2 \\ 6 & 0 & 9 \end{bmatrix} \to \mathbf{U} = \begin{bmatrix} 16 & 2 & 12 \\ 0 & 4 & 2 \\ 0 & 0 & 3 \end{bmatrix}$$

and thus we obtain the upper–triangular form of $\mathbf{J}$, namely

$$\mathbf{J}_u = \mathbf{G}_0\,\mathbf{U} = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} \begin{bmatrix} 16 & 2 & 12 \\ 0 & 4 & 2 \\ 0 & 0 & 3 \end{bmatrix} = \begin{bmatrix} 16d_1 & 2d_1 & 12d_1 \\ 0 & 4d_2 & 2d_2 \\ 0 & 0 & 3_3 \end{bmatrix} .$$

2.4 Triangularization with Permutations

Given an integer matrices $\mathbf{G}_{m\times n}$ and a $m \times m$ permutation matrix $\mathbf{P}$, the triangular form of the permuted matrix $\mathbf{G}_p = \mathbf{P}\,\mathbf{G}$ is evaluated, say $\mathbf{L} = \mathbf{G}_p\mathbf{E}$. Finally, the inverse permutation matrix is applied to $\mathbf{L}$ to get

$$\mathbf{L}_p == \mathbf{P}^{-1}\,\mathbf{L} .$$

This “permuted” triangularization will be used in Chap. 3 and in Chap. 6 to find the *primitive axis points* of a lattice.

2.4.1 *The Mathematica procedures “triangPU” and “triangPL”*

Usage

```
b = triangPU[a,vp,adj]
b = triangPL[a,vp,adj]
```

where

- `a` is the given integer matrix,
- `vp` is the permutation vector,
- `adj` is an *optional* argument with default value `adj = 1` , which provides the canonical form, while with `adj = 0` the form is not necessarily canonical,
- `b` is the permuted triangular form.

2.4.2 *Demos of “triangPU” and “triangPL”*

Given a matrix $\mathbf{A}$ of dimensions $m \times n$, the demo procedures `writetriangPU` and `writetriangPL` evaluates and write all the possible permuted triangular forms.

To get the demos the following is needed in `Mathematica`

1) define the integer matrix `a`,
2) define a permutation vector `p`,
3) write the statement `writesmithall[a]`.

Example 2.5. The first example of demo refers to square matrix 4×4. The statements for `Mathematica` are:

```
Get["demo1macro.m"]
lab="EL104"
initdemo1[lab]
g33= {{1,2,2},
        {0,3,0},
        {6,0,7}}

  vp={3,1,2}
writetriangPU[g33,0]
completedemo1
```

Hence `Mathematica` and TEX give

Triangularization: run $\boxed{\textbf{EL104}}$ (May 26, 2011)
Given an integer matrix **A** and a permutation vector **p**

$$\mathbf{A}=\begin{bmatrix}1&2&2\\0&3&0\\6&0&7\end{bmatrix}\qquad \mathbf{p}=[3\quad 1\quad 2]$$

the procedure `writetriangPU` finds the upper triangular form with permutation $\mathbf{U_p}$, that is,

$$\mathbf{U_p}=\mathbf{P}^{-1}[\mathbf{P}\,\mathbf{A}]\,\mathbf{E}\qquad \mathbf{U_p}=\mathbf{P}^{-1}\mathbf{U}$$

where **P** is the permutation matrix corresponding to **p**, **E** are unimodular, **U** is upper triangular.

---

matrix 3×3. `adj=1`

$$\mathbf{A}=\begin{bmatrix}1&2&2\\0&3&0\\6&0&7\end{bmatrix}\qquad \mathbf{p}=[3\quad 1\quad 2]\qquad \mathbf{P}=\begin{bmatrix}0&0&1\\1&0&0\\0&1&0\end{bmatrix}$$

permuted upper triangular form (evaluated with `triangPU`)

$$\mathbf{U}_p=\begin{bmatrix}0&1&0\\0&0&3\\5&1&3\end{bmatrix}$$

check (**E** evaluated with `rightUfactor` of **P A**)

$$\begin{aligned}&\mathbf{P}^{-1}[\mathbf{P}\,\mathbf{A}]\,\mathbf{E}\\ &=\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}\begin{bmatrix}0&0&1\\1&0&0\\0&1&0\end{bmatrix}\begin{bmatrix}1&2&2\\0&3&0\\6&0&7\end{bmatrix}\begin{bmatrix}2&-1&4\\0&0&1\\-1&1&-3\end{bmatrix}\\ &=\begin{bmatrix}0&1&0\\0&0&3\\5&1&3\end{bmatrix}\end{aligned}$$

Given a matrix **A** of dimensions $m \times n$, the procedures `writealltriangPU` and `writealltriangPL` evaluates and write all the permuted triangular forms

Example 2.6. With the `Mathematica` statements

```
Get["demo1macro.m"]
lab="EL106"
initdemo1[lab]
  b33={{1,2,3},
       {-5,2,12},
       {4,-2,8}}
writealltriangPU[b33]
completedemo1
```

Triangularization: run $\boxed{\textbf{EL106}}$ **(May 26, 2011)**

triangPL with all possible permutation. Given an integer matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ -5 & 2 & 12 \\ 4 & -2 & 8 \end{bmatrix}$$

the procedure `writealltriangPL` finds the generalized lower triangular forms $\mathbf{L_p}$ with all possible permutation vectors **p**

---

matrix 3×3**.** `adj=1`

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ -5 & 2 & 12 \\ 4 & -2 & 8 \end{bmatrix}$$

$$\mathbf{p} = [1 \quad 2 \quad 3] \qquad \mathbf{L}_p = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 3 & 0 \\ 36 & 16 & 74 \end{bmatrix}$$

$$\mathbf{p} = [1 \quad 3 \quad 2] \qquad \mathbf{L}_p = \begin{bmatrix} 1 & 0 & 0 \\ 22 & 42 & 111 \\ 0 & 2 & 0 \end{bmatrix}$$

$$\mathbf{p} = [2 \quad 1 \quad 3] \qquad \mathbf{L}_p = \begin{bmatrix} 1 & 3 & 0 \\ 1 & 0 & 0 \\ 36 & 18 & 74 \end{bmatrix}$$

$$\mathbf{p} = [2 \quad 3 \quad 1] \qquad \mathbf{L}_p = \begin{bmatrix} 106 & 99 & 111 \\ 1 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix}$$

$$\mathbf{p} = [3 \quad 1 \quad 2] \qquad \mathbf{L}_p = \begin{bmatrix} 0 & 1 & 0 \\ 42 & 22 & 111 \\ 2 & 0 & 0 \end{bmatrix}$$

$$\mathbf{p} = [3 \quad 2 \quad 1] \qquad \mathbf{L}_p = \begin{bmatrix} 99 & 106 & 111 \\ 0 & 1 & 0 \\ 2 & 0 & 0 \end{bmatrix}$$

2.5 Smith Diagonalization

From Section 16.6 of the *Unified Signal Theory* we recall:

Theorem 2.2. *An integer matrix* $\mathbf{H}$ *can be decomposed in the form*

$$\underset{m\times n}{\mathbf{H}} = \underset{m\times m}{\mathbf{E}_1}\ \underset{m\times n}{\boldsymbol{\Delta}}\ \underset{n\times n}{\mathbf{E}_2}$$

where $\boldsymbol{\Delta}$ *is diagonal and* $\mathbf{E}_1$, $\mathbf{E}_2$ *are unimodular.*

To get this decomposition elementary operations on both rows an columns are needed. In the case of nonsingular square matrices it is possible to get a *canonical* form, where the diagonal entries are positive.

2.5.1 Diagonalization by hand

Example 2.7. We reconsider Example 2.1, where for the matrix

$$\mathbf{H} = \begin{bmatrix} 3 & -1 & 4 \\ 2 & 1 & -3 \\ 0 & 2 & -1 \end{bmatrix} \quad \to \quad \mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 5 & 7 & 29 \end{bmatrix}$$

we have already obtained the lower triangular form $\mathbf{L} = \mathbf{H}\mathbf{E}_2$ and $\mathbf{E}_2$ can be obtained as $\mathbf{H}^{-1}\mathbf{L}$, namely

$$\mathbf{E}_2 = \mathbf{H}^{-1}\mathbf{L} \Rightarrow \begin{bmatrix} 3 & -1 & 4 \\ 2 & 1 & -3 \\ 0 & 2 & -1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 5 & 7 & 29 \end{bmatrix} = \begin{bmatrix} 3 & -1 & 4 \\ 2 & 1 & -3 \\ -1 & 0 & 0 \end{bmatrix}$$

Now we work on the rows of $\mathbf{L}$, specifically

- summing to the third row the second multiplied by -7,
- summing to the third row the first multiplied by -5,

we gets

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 5 & 7 & 9 \end{bmatrix} \to \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 5 & 0 & 29 \end{bmatrix} \to \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 29 \end{bmatrix} = \boldsymbol{\Delta}\ .$$

Hence we get $\mathbf{E}_1$ as $\mathbf{L}\boldsymbol{\Delta}^{-1}$.

2.5.2 *The Mathematica procedure "smith"*

Usage

`{g, diag, h} = smith[a, adj]`

where

- `a` is the given integer matrix,
- `adj` is an *optional* argument with default value `adj = 1`, which provides the canonical form, while with `adj = 0` **U** and **L** are not necessarily in canonical form,
- `g` and `h` are unimodular matrices,
- `diag` is a diagonal matrix.

Hence we have the two possible forms

```
{g, diag, h} = smith[a]
{g, diag, h} = smith[a,0].
```

2.5.3 *Demo of "smith"*

Given a matrix **A** of dimensions $m \times n$, the demo procedure `writesmith` evaluates and writes the Hermite decomposition.

To get the demos the following is needed in `Mathematica`

1) define the integer matrix `a`
2) write the statement `writesmith[a]`.

Example 2.8. The first example of demo refers to a square matrix 4×4. The program for `Mathematica` is

```
Get["demo1macro.m"]
lab="EL108"
initdemo1[lab]
g33= {{1,2,2},
        {0,3,0},
        {6,0,7}}

  vp={3,1,2}
writesmith[g33,0]
completedemo1
```

Hence TEX gives

Diagonaliazation: run $\boxed{\textbf{EL108}}$ **(May 26, 2011)**
Given an integer matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ -5 & 2 & 12 \\ 4 & -2 & 8 \end{bmatrix}$$

the procedure `writesmith` writes the Smith decomposition, that is,

$$\mathbf{A} = \mathbf{E}\,\Delta\,\mathbf{F}$$

with **E** unimodular, Δ diagonal and **F** unimodular

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ -5 & 2 & 12 \\ 4 & -2 & 8 \end{bmatrix}$$

$$\mathbf{E}\,\Delta\,\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 \\ -5 & 3 & 1 \\ 4 & 16 & 5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 222 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 0 & -70 & -139 \\ 0 & 1 & 2 \end{bmatrix}$$

2.6 Triangularization and Diagonalization of Rectangular Matrices

The above procedures of triangularization and diagonalization, namely

- `hermite(a,adj)`,
- `triangU(a,adj)`,
- `triangL(a,adj)`,
- `smith(a,adj)`

hold also with rectangular matrices, in agreement with Theorem 2.1 and Theorem 2.2, although they were applied and illustrated for square matrices (the case of main interest for lattices). In this section the procedures are illustrated for rectangular matrices using the corresponding demos.

An important remark is that in some cases the triangularization should be intended in a generalized sense and not in the strict sense of the definition given in Sect. 2.5. What is true is that in any case in the rectangular matrix we will find a square triangular submatrix. On the other hand the diagonalization works in the strict sense.

Example 2.9. We use the demo $\boxed{\text{writehermiteall}}$ with a 3×7 matrix.

```
Get["demo1macro.m"]
lab="EL114"
initdemo1[lab]
c37={{0,2,3,5,6,17,8},
    {3,0,2,2,3,7,8},
```

```
      {4,-1,3,1,14,7,8}}
  writehermiteall[c37,0]
  completedemo1
```

Triangularization: run $\boxed{\textbf{EL114}}$ **(May 26, 2011)**

Given an integer matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 2 & 3 & 5 & 6 & 17 & 8 \\ 3 & 0 & 2 & 2 & 3 & 7 & 8 \\ 4 & -1 & 3 & 1 & 14 & 7 & 8 \end{bmatrix}$$

the procedure `writehermiteall` writes the decompositions

$$\mathbf{AE} = \mathbf{L} \qquad \mathbf{AE} = \mathbf{U} \qquad \mathbf{EA} = \mathbf{L} \qquad \mathbf{EA} = \mathbf{U}$$

where **E** are unimodular, **L** are lower triangular, **U** are upper triangular.
If `adj = 1` the triangular form is "adjusted" to have a canonical form.

---

matrix 3×7. `adj=1` $\mathbf{AE} = \mathbf{L}$

$$\begin{bmatrix} 0 & 2 & 3 & 5 & 6 & 17 & 8 \\ 3 & 0 & 2 & 2 & 3 & 7 & 8 \\ 4 & -1 & 3 & 1 & 14 & 7 & 8 \end{bmatrix} \begin{bmatrix} 2 & -1 & 0 & -4 & -9 & -27 & -8 \\ 11 & -8 & 1 & -20 & -56 & -138 & -32 \\ 3 & -3 & 1 & -5 & -23 & -37 & -8 \\ -6 & 5 & -1 & 11 & 35 & 74 & 16 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

matrix 3×7. `adj=1` $\mathbf{AE} = \mathbf{U}$

$$\begin{bmatrix} 0 & 2 & 3 & 5 & 6 & 17 & 8 \\ 3 & 0 & 2 & 2 & 3 & 7 & 8 \\ 4 & -1 & 3 & 1 & 14 & 7 & 8 \end{bmatrix} \begin{bmatrix} 56 & 69 & 71 & 4 & -2 & -1 & -4 \\ 288 & 342 & 344 & 20 & -9 & -8 & -19 \\ 72 & 83 & 77 & 5 & -2 & -3 & -4 \\ -160 & -190 & -185 & -11 & 5 & 5 & 10 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} =$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

matrix 3×7. `adj=1` $\mathbf{EA} = \mathbf{L}$

$$\begin{bmatrix} 0 & -1 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 2 & 3 & 5 & 6 & 17 & 8 \\ 3 & 0 & 2 & 2 & 3 & 7 & 8 \\ 4 & -1 & 3 & 1 & 14 & 7 & 8 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 & 11 & 0 & 0 \\ -3 & 2 & 1 & 3 & 3 & 10 & 0 \\ 3 & 0 & 2 & 2 & 3 & 7 & 8 \end{bmatrix}$$

matrix 3×7. `adj=1` $\mathbf{EA} = \mathbf{U}$

$$\begin{bmatrix} 2 & -5 & 4 \\ 2 & -4 & 3 \\ 3 & -8 & 6 \end{bmatrix} \begin{bmatrix} 0 & 2 & 3 & 5 & 6 & 17 & 8 \\ 3 & 0 & 2 & 2 & 3 & 7 & 8 \\ 4 & -1 & 3 & 1 & 14 & 7 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 8 & 4 & 53 & 27 & 8 \\ 0 & 1 & 7 & 5 & 42 & 27 & 8 \\ 0 & 0 & 11 & 5 & 78 & 37 & 8 \end{bmatrix}$$

In this example we realize that the triangularizations $\mathbf{AE} = \mathbf{L}$ and $\mathbf{EA} = \mathbf{U}$ hold in the strict sense, whereas the triangularizations $\mathbf{AE} = \mathbf{U}$ and $\mathbf{EA} = \mathbf{L}$ hold in the generalized sense (they contain a 3×3 triangular matrix in a wrong place).

Example 2.10. We use the demo $\boxed{\text{writehermiteall}}$ with 7×3 matrix, obtained by transposition of the previous matrix. The program for `Mathematica` is

```
Get["demo1macro.m"]
lab="EL122"
initdemo1[lab]
c37={{0,2,3,5,6,17,8},
    {3,0,2,2,3,7,8},
    {4,-1,3,1,14,7,8}}
writehermiteall[Transpose[c37],0]
completedemo1
```

Hence TEX gives

Triangularization: run $\boxed{\textbf{EL122}}$ **(May 26, 2011)**
Given an integer matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 3 & 4 \\ 2 & 0 & -1 \\ 3 & 2 & 3 \\ 5 & 2 & 1 \\ 6 & 3 & 14 \\ 17 & 7 & 7 \\ 8 & 8 & 8 \end{bmatrix}$$

the procedure `writehermiteall` writes the decompositions

$$\mathbf{AE} = \mathbf{L} \qquad \mathbf{AE} = \mathbf{U} \qquad \mathbf{EA} = \mathbf{L} \qquad \mathbf{EA} = \mathbf{U}$$

where $\mathbf{E}$ are unimodular, $\mathbf{L}$ are lower triangular, $\mathbf{U}$ are upper triangular.
If `adj = 1` the triangular form is "adjusted" to have a canonical form.

---

matrix 7×3. `adj=1` $\mathbf{AE} = \mathbf{L}$

$$\begin{bmatrix} 0 & 3 & 4 \\ 2 & 0 & -1 \\ 3 & 2 & 3 \\ 5 & 2 & 1 \\ 6 & 3 & 14 \\ 17 & 7 & 7 \\ 8 & 8 & 8 \end{bmatrix} \begin{bmatrix} 2 & 2 & 3 \\ -5 & -4 & -8 \\ 4 & 3 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 8 & 7 & 11 \\ 4 & 5 & 5 \\ 53 & 42 & 78 \\ 27 & 27 & 37 \\ 8 & 8 & 8 \end{bmatrix}$$

matrix 7×3. `adj=1` $\mathbf{AE} = \mathbf{U}$

$$\begin{bmatrix} 0 & 3 & 4 \\ 2 & 0 & -1 \\ 3 & 2 & 3 \\ 5 & 2 & 1 \\ 6 & 3 & 14 \\ 17 & 7 & 7 \\ 8 & 8 & 8 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & -1 & 1 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & -3 & 3 \\ -1 & 2 & 0 \\ 1 & 1 & 2 \\ -1 & 3 & 2 \\ 11 & 3 & 3 \\ 0 & 10 & 7 \\ 0 & 0 & 8 \end{bmatrix}$$

matrix 7×3. `adj=1` $\mathbf{EA} = \mathbf{L}$

$$\begin{bmatrix} -2 & -9 & -2 & 5 & 0 & 0 & 0 \\ -1 & -8 & -3 & 5 & 0 & 0 & 0 \\ -4 & -19 & -4 & 10 & 0 & 0 & 0 \\ 56 & 288 & 72 & -160 & 0 & 0 & 1 \\ 69 & 342 & 83 & -190 & 0 & 1 & 0 \\ 71 & 344 & 77 & -185 & 1 & 0 & 0 \\ 4 & 20 & 5 & -11 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 3 & 4 \\ 2 & 0 & -1 \\ 3 & 2 & 3 \\ 5 & 2 & 1 \\ 6 & 3 & 14 \\ 17 & 7 & 7 \\ 8 & 8 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

matrix 7×3. `adj=1` $\mathbf{EA} = \mathbf{U}$

$$\begin{bmatrix} 2 & 11 & 3 & -6 & 0 & 0 & 0 \\ -1 & -8 & -3 & 5 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 & 0 \\ -4 & -20 & -5 & 11 & 0 & 0 & 0 \\ -9 & -56 & -23 & 35 & 1 & 0 & 0 \\ -27 & -138 & -37 & 74 & 0 & 1 & 0 \\ -8 & -32 & -8 & 16 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 3 & 4 \\ 2 & 0 & -1 \\ 3 & 2 & 3 \\ 5 & 2 & 1 \\ 6 & 3 & 14 \\ 17 & 7 & 7 \\ 8 & 8 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Also in this example we see that the triangularizations $\mathbf{AE} = \mathbf{L}$ and $\mathbf{EA} = \mathbf{U}$ hold in the strict sense, whereas the triangularizations $\mathbf{AE} = \mathbf{U}$ and $\mathbf{EA} = \mathbf{L}$ hold in the generalized sense (they contain a 3×3 triangular matrix in a wrong place).

Example 2.11. We use the demo $\boxed{\text{writesmith}}$ twice with a 3×7 matrix. The program for `Mathematica` is

```
lab="EL124"
Get["demo1macro.m"]
initdemo1[lab]
c37={{0,2,3,5,6,17,8},
    {3,0,2,2,3,7,8},
    {4,-1,3,1,14,7,8}}
writesmith[c37,0]
completedemo1
```

Hence TEX gives

Diagonaliazation: run $\boxed{\textbf{EL124}}$ **(May 26, 2011)**
Given an integer matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 2 & 3 & 5 & 6 & 17 & 8 \\ 3 & 0 & 2 & 2 & 3 & 7 & 8 \\ 4 & -1 & 3 & 1 & 14 & 7 & 8 \end{bmatrix}$$

the procedure `writesmith` writes the Smith decomposition, that is,

$$\mathbf{A} = \mathbf{E}\,\Delta\,\mathbf{F}$$

with $\mathbf{E}$ unimodular, Δ diagonal and $\mathbf{F}$ unimodular

---

$$\mathbf{A} = \begin{bmatrix} 0 & 2 & 3 & 5 & 6 & 17 & 8 \\ 3 & 0 & 2 & 2 & 3 & 7 & 8 \\ 4 & -1 & 3 & 1 & 14 & 7 & 8 \end{bmatrix}$$

$$\mathbf{E}\,\Delta\,\mathbf{F} =$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & -1 & 0 \\ 4 & -5 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 2 & 3 & 5 & 6 & 17 & 8 \\ -3 & 4 & 4 & 8 & 9 & 27 & 8 \\ 11 & -11 & -11 & -21 & -35 & -74 & -16 \\ -1 & 1 & 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Example 2.12. We use the demo $\boxed{\text{writesmith}}$ with the transpose of the previous matrix `c37`. The program for `Mathematica` is

```
lab="EL126"
Get["demo1macro.m"]
initdemo1[lab]
c73=Transpose[c37]
writesmith[Transpose[c37],0]
completedemo1
```

Hence TEX gives

Diagonaliazation: run $\boxed{\textbf{EL126}}$ **(May 26, 2011)**
Given an integer matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 3 & 4 \\ 2 & 0 & -1 \\ 3 & 2 & 3 \\ 5 & 2 & 1 \\ 6 & 3 & 14 \\ 17 & 7 & 7 \\ 8 & 8 & 8 \end{bmatrix}$$

the procedure `writesmith` writes the Smith decomposition, that is,

$$\mathbf{A} = \mathbf{E}\,\Delta\,\mathbf{F}$$

with $\mathbf{E}$ unimodular, Δ diagonal and $\mathbf{F}$ unimodular

---

$$\mathbf{A} = \begin{bmatrix} 0 & 3 & 4 \\ 2 & 0 & -1 \\ 3 & 2 & 3 \\ 5 & 2 & 1 \\ 6 & 3 & 14 \\ 17 & 7 & 7 \\ 8 & 8 & 8 \end{bmatrix}$$

$$\mathbf{E}\,\Delta\,\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 11 & 2 & 0 & 0 & 0 \\ -1 & 0 & 5 & 1 & 0 & 0 & 0 \\ 11 & -36 & 78 & 0 & 1 & 0 & 0 \\ 0 & -10 & 37 & 0 & 0 & 1 & 0 \\ 0 & 0 & 8 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 3 & 4 \\ 2 & 3 & 3 \\ 1 & 1 & 1 \end{bmatrix}$$

In both examples we realize that the diagonalization of the rectangular matrices hold in the strict sense.

2.7 Other Demos with rectangular matrices

Hermite decomposition *(May 26, 2011)*

Given an integer matrix **A**, it is shown the decompositions

$$\mathbf{A\,E} = \mathbf{L} \qquad \mathbf{A\,E} = \mathbf{U} \qquad \mathbf{E\,A} = \mathbf{L} \qquad \mathbf{E\,A} = \mathbf{U}$$

where **E** are unimodular, **L** are lower triangular, **U** are upper triangular.
If `adj = 1` the triangular form is "adjusted" to have a canonical form

matrix 3×5. `adj=1` $\mathbf{A\,E} = \mathbf{L}$

$$\begin{bmatrix} 3 & -2 & 9 & 4 & 11 \\ 3 & 3 & 0 & 12 & -8 \\ 10 & 1 & 1 & -5 & -7 \end{bmatrix} \begin{bmatrix} -27359 & 4571 & -6 & 93 & 13499 \\ 116763 & -19508 & 26 & -397 & -57611 \\ 45010 & -7520 & 10 & -153 & -22208 \\ -22355 & 3735 & -5 & 76 & 11030 \\ -6 & 1 & 0 & 0 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

matrix 3×5. `adj=1` $\mathbf{A\,E} = \mathbf{U}$

$$\begin{bmatrix} 3 & -2 & 9 & 4 & 11 \\ 3 & 3 & 0 & 12 & -8 \\ 10 & 1 & 1 & -5 & -7 \end{bmatrix} \begin{bmatrix} -1474 & 93 & 11 & -451 & 1375 \\ 6306 & -397 & -47 & 1930 & -5883 \\ 2425 & -153 & -18 & 742 & -2262 \\ -1206 & 76 & 9 & -369 & 1125 \\ 3 & 0 & 0 & 1 & -3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

matrix 3×5. `adj=1` $\mathbf{E\,A} = \mathbf{L}$

$$\begin{bmatrix} 124 & 27 & 164 \\ 87 & 19 & 115 \\ 115 & 25 & 152 \end{bmatrix} \begin{bmatrix} 3 & -2 & 9 & 4 & 11 \\ 3 & 3 & 0 & 12 & -8 \\ 10 & 1 & 1 & -5 & -7 \end{bmatrix} = \begin{bmatrix} 2093 & -3 & 1280 & 0 & 0 \\ 1468 & -2 & 898 & 1 & 0 \\ 1940 & -3 & 1187 & 0 & 1 \end{bmatrix}$$

matrix 3×5. `adj=1` $\mathbf{E\,A} = \mathbf{U}$

$$\begin{bmatrix} 20 & 17 & -11 \\ 16 & 14 & -9 \\ 27 & 23 & -15 \end{bmatrix} \begin{bmatrix} 3 & -2 & 9 & 4 & 11 \\ 3 & 3 & 0 & 12 & -8 \\ 10 & 1 & 1 & -5 & -7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 169 & 339 & 161 \\ 0 & 1 & 135 & 277 & 127 \\ 0 & 0 & 228 & 459 & 218 \end{bmatrix}$$

Smith diagonalization *(May 26, 2011)*

Given an integer matrix **A** the demo evaluates and writes the Smith decomposition, that is,

$$\mathbf{A} = \mathbf{E}\,\Delta\,\mathbf{F}$$

with **E** unimodular, Δ diagonal and **F** unimodular

$$\mathbf{A} = \begin{bmatrix} 3 & -2 & 9 & 4 & 11 \\ 3 & 3 & 0 & 12 & -8 \\ 10 & 1 & 1 & -5 & -7 \end{bmatrix}$$

$$\mathbf{E}\,\Delta\,\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 \\ 6 & -1 & 0 \\ 11 & -747 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 3 & -2 & 9 & 4 & 11 \\ 15 & -15 & 54 & 12 & 74 \\ -11182 & 11182 & -40240 & -8915 & -55150 \\ 10 & -10 & 35 & 6 & 0 \\ 5 & -5 & 18 & 4 & 25 \end{bmatrix}$$

Hermite decomposition *(May 26, 2011)*

Given an integer matrix **A**, it is shown the decompositions

$$\mathbf{A}\,\mathbf{E} = \mathbf{L} \qquad \mathbf{A}\,\mathbf{E} = \mathbf{U} \qquad \mathbf{E}\,\mathbf{A} = \mathbf{L} \qquad \mathbf{E}\,\mathbf{A} = \mathbf{U}$$

where **E** are unimodular, **L** are lower triangular, **U** are upper triangular.
If `adj = 1` the triangular form is “adjusted” to have a canonical form

matrix 3×8**.** `adj=1` $\mathbf{A}\,\mathbf{E} = \mathbf{L}$

$$\begin{bmatrix} 6 & 2 & 3 & 3 & 6 & 7 & 8 & -1 \\ 3 & 0 & 2 & 16 & 3 & 7 & 8 & -3 \\ 4 & -1 & 0 & 1 & 4 & 7 & 8 & -2 \end{bmatrix} \begin{bmatrix} 20 & -9 & -2 & 2 & -1 & -21 & -72 & -11 \\ -46 & 20 & 5 & 9 & 0 & 49 & 168 & 24 \\ 33 & -14 & -4 & -11 & 0 & -35 & -120 & -17 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -18 & 8 & 2 & 0 & 0 & 19 & 64 & 10 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

matrix 3×8**.** `adj=1` $\mathbf{A}\,\mathbf{E} = \mathbf{U}$

$$\begin{bmatrix} 6 & 2 & 3 & 3 & 6 & 7 & 8 & -1 \\ 3 & 0 & 2 & 16 & 3 & 7 & 8 & -3 \\ 4 & -1 & 0 & 1 & 4 & 7 & 8 & -2 \end{bmatrix} \begin{bmatrix} 31 & -72 & -21 & -1 & 2 & -1 & 12 & -2 \\ -74 & 168 & 49 & 0 & 9 & 3 & -29 & 5 \\ 53 & -120 & -35 & 0 & -11 & -2 & 21 & -4 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -28 & 64 & 19 & 0 & 0 & 1 & -11 & 2 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

matrix 3×8**.** `adj=1` $\mathbf{E}\,\mathbf{A} = \mathbf{L}$

$$\begin{bmatrix} 1 & 1 & -2 \\ 2 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 6 & 2 & 3 & 3 & 6 & 7 & 8 & -1 \\ 3 & 0 & 2 & 16 & 3 & 7 & 8 & -3 \\ 4 & -1 & 0 & 1 & 4 & 7 & 8 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 5 & 17 & 1 & 0 & 0 & 0 \\ 8 & 5 & 6 & 5 & 8 & 7 & 8 & 0 \\ 2 & 3 & 3 & 2 & 2 & 0 & 0 & 1 \end{bmatrix}$$

matrix 3×8**.** `adj=1` $\mathbf{E}\,\mathbf{A} = \mathbf{U}$

$$\begin{bmatrix} -1 & 5 & -2 \\ -1 & 6 & -3 \\ -3 & 14 & -6 \end{bmatrix} \begin{bmatrix} 6 & 2 & 3 & 3 & 6 & 7 & 8 & -1 \\ 3 & 0 & 2 & 16 & 3 & 7 & 8 & -3 \\ 4 & -1 & 0 & 1 & 4 & 7 & 8 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 7 & 75 & 1 & 14 & 16 & -10 \\ 0 & 1 & 9 & 90 & 0 & 14 & 16 & -11 \\ 0 & 0 & 19 & 209 & 0 & 35 & 40 & -27 \end{bmatrix}$$

Smith diagonalization *(May 26, 2011)*

Given an integer matrix **A** the demo evaluates and writes the Smith decomposition, that is,

$$\mathbf{A} = \mathbf{E}\,\Delta\,\mathbf{F}$$

with **E** unimodular, Δ diagonal and **F** unimodular

$$\mathbf{A} = \begin{bmatrix} 6 & 2 & 3 & 3 & 6 & 7 & 8 & -1 \\ 3 & 0 & 2 & 16 & 3 & 7 & 8 & -3 \\ 4 & -1 & 0 & 1 & 4 & 7 & 8 & -2 \end{bmatrix}$$

$$\mathbf{E}\,\Delta\,\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & -1 & 0 \\ 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 6 & 2 & 3 & 3 & 6 & 7 & 8 & -1 \\ 9 & 4 & 4 & -10 & 9 & 7 & 8 & 1 \\ -38 & -19 & -19 & 38 & -38 & -28 & -32 & -5 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 4 & 2 & 2 & -4 & 4 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Hermite decomposition *(May 26, 2011)*

Given an integer matrix **A**, it is shown the decompositions

$$\mathbf{A}\mathbf{E} = \mathbf{L} \qquad \mathbf{A}\mathbf{E} = \mathbf{U} \qquad \mathbf{E}\mathbf{A} = \mathbf{L} \qquad \mathbf{E}\mathbf{A} = \mathbf{U}$$

where **E** are unimodular, **L** are lower triangular, **U** are upper triangular.
If `adj = 1` the triangular form is “adjusted” to have a canonical form

matrix 8×3**.** `adj=1` $\mathbf{A}\mathbf{E} = \mathbf{L}$

$$\begin{bmatrix} 6 & 3 & 4 \\ 2 & 0 & -1 \\ 3 & 2 & 0 \\ 3 & 16 & 1 \\ 6 & 3 & 4 \\ 7 & 7 & 7 \\ 8 & 8 & 8 \\ -1 & -3 & -2 \end{bmatrix} \begin{bmatrix} -1 & -1 & -3 \\ 5 & 6 & 14 \\ -2 & -3 & -6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 7 & 9 & 19 \\ 75 & 90 & 209 \\ 1 & 0 & 0 \\ 14 & 14 & 35 \\ 16 & 16 & 40 \\ -10 & -11 & -27 \end{bmatrix}$$

matrix 8×3**.** `adj=1` $\mathbf{A}\mathbf{E} = \mathbf{U}$

$$\begin{bmatrix} 6 & 3 & 4 \\ 2 & 0 & -1 \\ 3 & 2 & 0 \\ 3 & 16 & 1 \\ 6 & 3 & 4 \\ 7 & 7 & 7 \\ 8 & 8 & 8 \\ -1 & -3 & -2 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 1 & 0 & 0 \\ -2 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 8 & 2 \\ 4 & 5 & 3 \\ 5 & 6 & 3 \\ 17 & 5 & 2 \\ 1 & 8 & 2 \\ 0 & 7 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

matrix 8×3**.** `adj=1` $\mathbf{E}\mathbf{A} = \mathbf{L}$

$$\begin{bmatrix} -1 & 3 & -2 & 0 & 0 & 1 & 0 & 0 \\ 12 & -29 & 21 & 0 & 0 & -11 & 0 & 0 \\ -2 & 5 & -4 & 0 & 0 & 2 & 0 & 0 \\ 31 & -74 & 53 & 0 & 0 & -28 & 0 & 1 \\ -72 & 168 & -120 & 0 & 0 & 64 & 1 & 0 \\ -21 & 49 & -35 & 0 & 0 & 19 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & 9 & -11 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 6 & 3 & 4 \\ 2 & 0 & -1 \\ 3 & 2 & 0 \\ 3 & 16 & 1 \\ 6 & 3 & 4 \\ 7 & 7 & 7 \\ 8 & 8 & 8 \\ -1 & -3 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

matrix 8×3. `adj=1` $\mathbf{E}\,\mathbf{A} = \mathbf{U}$

$$\begin{bmatrix} 20 & -46 & 33 & 0 & 0 & -18 & 0 & 0 \\ -9 & 20 & -14 & 0 & 0 & 8 & 0 & 0 \\ -2 & 5 & -4 & 0 & 0 & 2 & 0 & 0 \\ 2 & 9 & -11 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -21 & 49 & -35 & 0 & 0 & 19 & 0 & 0 \\ -72 & 168 & -120 & 0 & 0 & 64 & 1 & 0 \\ -11 & 24 & -17 & 0 & 0 & 10 & 0 & 1 \end{bmatrix} \begin{bmatrix} 6 & 3 & 4 \\ 2 & 0 & -1 \\ 3 & 2 & 0 \\ 3 & 16 & 1 \\ 6 & 3 & 4 \\ 7 & 7 & 7 \\ 8 & 8 & 8 \\ -1 & -3 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Smith diagonalization ***(May 26, 2011)***

Given an integer matrix **A** the demo evaluates and writes the Smith decomposition, that is,

$$\mathbf{A} = \mathbf{E}\,\Delta\,\mathbf{F}$$

with **E** unimodular, Δ diagonal and **F** unimodular

$$\mathbf{A} = \begin{bmatrix} 0 & 2 & 3 & 5 & 6 & 17 & 8 \\ 3 & 0 & 2 & 2 & 3 & 7 & 8 \\ 4 & -1 & 3 & 1 & 14 & 7 & 8 \end{bmatrix}$$

$$\mathbf{E}\,\Delta\,\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & -1 & 0 \\ 4 & -5 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 2 & 3 & 5 & 6 & 17 & 8 \\ -3 & 4 & 4 & 8 & 9 & 27 & 8 \\ 11 & -11 & -11 & -21 & -35 & -74 & -16 \\ -1 & 1 & 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Hermite decomposition ***(May 26, 2011)***

Given an integer matrix **A**, it is shown the decompositions

$$\mathbf{A}\,\mathbf{E} = \mathbf{L} \qquad \mathbf{A}\,\mathbf{E} = \mathbf{U} \qquad \mathbf{E}\,\mathbf{A} = \mathbf{L} \qquad \mathbf{E}\,\mathbf{A} = \mathbf{U}$$

where **E** are unimodular, **L** are lower triangular, **U** are upper triangular.
If `adj = 1` the triangular form is “adjusted” to have a canonical form

matrix 7×3. `adj=1` $\mathbf{A}\,\mathbf{E} = \mathbf{L}$

$$\begin{bmatrix} 0 & 3 & 4 \\ 2 & 0 & -1 \\ 3 & 2 & 3 \\ 5 & 2 & 1 \\ 6 & 3 & 14 \\ 17 & 7 & 7 \\ 8 & 8 & 8 \end{bmatrix} \begin{bmatrix} 2 & 2 & 3 \\ -5 & -4 & -8 \\ 4 & 3 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 8 & 7 & 11 \\ 4 & 5 & 5 \\ 53 & 42 & 78 \\ 27 & 27 & 37 \\ 8 & 8 & 8 \end{bmatrix}$$

matrix 7×3**.** `adj=1` $\mathbf{A}\mathbf{E} = \mathbf{U}$

$$\begin{bmatrix} 0 & 3 & 4 \\ 2 & 0 & -1 \\ 3 & 2 & 3 \\ 5 & 2 & 1 \\ 6 & 3 & 14 \\ 17 & 7 & 7 \\ 8 & 8 & 8 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & -1 & 1 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & -3 & 3 \\ -1 & 2 & 0 \\ 1 & 1 & 2 \\ -1 & 3 & 2 \\ 11 & 3 & 3 \\ 0 & 10 & 7 \\ 0 & 0 & 8 \end{bmatrix}$$

matrix 7×3**.** `adj=1` $\mathbf{E}\mathbf{A} = \mathbf{L}$

$$\begin{bmatrix} -2 & -9 & -2 & 5 & 0 & 0 & 0 \\ -1 & -8 & -3 & 5 & 0 & 0 & 0 \\ -4 & -19 & -4 & 10 & 0 & 0 & 0 \\ 56 & 288 & 72 & -160 & 0 & 0 & 1 \\ 69 & 342 & 83 & -190 & 0 & 1 & 0 \\ 71 & 344 & 77 & -185 & 1 & 0 & 0 \\ 4 & 20 & 5 & -11 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 3 & 4 \\ 2 & 0 & -1 \\ 3 & 2 & 3 \\ 5 & 2 & 1 \\ 6 & 3 & 14 \\ 17 & 7 & 7 \\ 8 & 8 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

matrix 7×3**.** `adj=1` $\mathbf{E}\mathbf{A} = \mathbf{U}$

$$\begin{bmatrix} 2 & 11 & 3 & -6 & 0 & 0 & 0 \\ -1 & -8 & -3 & 5 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 & 0 \\ -4 & -20 & -5 & 11 & 0 & 0 & 0 \\ -9 & -56 & -23 & 35 & 1 & 0 & 0 \\ -27 & -138 & -37 & 74 & 0 & 1 & 0 \\ -8 & -32 & -8 & 16 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 3 & 4 \\ 2 & 0 & -1 \\ 3 & 2 & 3 \\ 5 & 2 & 1 \\ 6 & 3 & 14 \\ 17 & 7 & 7 \\ 8 & 8 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Smith diagonalization ***(May 26, 2011)***

Given an integer matrix **A** the demo evaluates and writes the Smith decomposition, that is,

$$\mathbf{A} = \mathbf{E}\,\Delta\,\mathbf{F}$$

with **E** unimodular, Δ diagonal and **F** unimodular

$$\mathbf{A} = \begin{bmatrix} 0 & 3 & 4 \\ 2 & 0 & -1 \\ 3 & 2 & 3 \\ 5 & 2 & 1 \\ 6 & 3 & 14 \\ 17 & 7 & 7 \\ 8 & 8 & 8 \end{bmatrix}$$

$$\mathbf{E}\,\Delta\,\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 11 & 2 & 0 & 0 & 0 \\ -1 & 0 & 5 & 1 & 0 & 0 & 0 \\ 11 & -36 & 78 & 0 & 1 & 0 & 0 \\ 0 & -10 & 37 & 0 & 0 & 1 & 0 \\ 0 & 0 & 8 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 3 & 4 \\ 2 & 3 & 3 \\ 1 & 1 & 1 \end{bmatrix}$$

Hermite decomposition ***(May 26, 2011)***

Given an integer matrix **A**, it is shown the decompositions

$$\mathbf{A}\mathbf{E} = \mathbf{L} \qquad \mathbf{A}\mathbf{E} = \mathbf{U} \qquad \mathbf{E}\mathbf{A} = \mathbf{L} \qquad \mathbf{E}\mathbf{A} = \mathbf{U}$$

where **E** are unimodular, **L** are lower triangular, **U** are upper triangular.
If `adj = 1` the triangular form is "adjusted" to have a canonical form

matrix 3×7**.** `adj=1` $\mathbf{A}\mathbf{E} = \mathbf{L}$

$$\begin{bmatrix} 0 & 2 & 3 & 5 & 6 & 17 & 8 \\ 3 & 0 & 2 & 2 & 3 & 7 & 8 \\ 4 & -1 & 3 & 1 & 14 & 7 & 8 \end{bmatrix} \begin{bmatrix} 2 & -1 & 0 & -4 & -9 & -27 & -8 \\ 11 & -8 & 1 & -20 & -56 & -138 & -32 \\ 3 & -3 & 1 & -5 & -23 & -37 & -8 \\ -6 & 5 & -1 & 11 & 35 & 74 & 16 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

matrix 3×7**.** `adj=1` $\mathbf{A}\mathbf{E} = \mathbf{U}$

$$\begin{bmatrix} 0 & 2 & 3 & 5 & 6 & 17 & 8 \\ 3 & 0 & 2 & 2 & 3 & 7 & 8 \\ 4 & -1 & 3 & 1 & 14 & 7 & 8 \end{bmatrix} \begin{bmatrix} 56 & 69 & 71 & 4 & -2 & -1 & -4 \\ 288 & 342 & 344 & 20 & -9 & -8 & -19 \\ 72 & 83 & 77 & 5 & -2 & -3 & -4 \\ -160 & -190 & -185 & -11 & 5 & 5 & 10 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

matrix 3×7**.** `adj=1` $\mathbf{E}\mathbf{A} = \mathbf{L}$

$$\begin{bmatrix} 0 & -1 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 2 & 3 & 5 & 6 & 17 & 8 \\ 3 & 0 & 2 & 2 & 3 & 7 & 8 \\ 4 & -1 & 3 & 1 & 14 & 7 & 8 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 & 11 & 0 & 0 \\ -3 & 2 & 1 & 3 & 3 & 10 & 0 \\ 3 & 0 & 2 & 2 & 3 & 7 & 8 \end{bmatrix}$$

matrix 3×7**.** `adj=1` $\mathbf{E}\mathbf{A} = \mathbf{U}$

$$\begin{bmatrix} 2 & -5 & 4 \\ 2 & -4 & 3 \\ 3 & -8 & 6 \end{bmatrix} \begin{bmatrix} 0 & 2 & 3 & 5 & 6 & 17 & 8 \\ 3 & 0 & 2 & 2 & 3 & 7 & 8 \\ 4 & -1 & 3 & 1 & 14 & 7 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 8 & 4 & 53 & 27 & 8 \\ 0 & 1 & 7 & 5 & 42 & 27 & 8 \\ 0 & 0 & 11 & 5 & 78 & 37 & 8 \end{bmatrix}$$

Smith diagonalization ***(May 26, 2011)***

Given an integer matrix **A** the demo evaluates and writes the Smith decomposition, that is,

$$\mathbf{A} = \mathbf{E}\,\Delta\,\mathbf{F}$$

with **E** unimodular, Δ diagonal and **F** unimodular

$$\mathbf{A} = \begin{bmatrix} 0 & 2 & 3 & 5 & 6 & 17 & 8 \\ 3 & 0 & 2 & 2 & 3 & 7 & 8 \\ 4 & -1 & 3 & 1 & 14 & 7 & 8 \end{bmatrix}$$

$$\mathbf{E}\,\Delta\,\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & -1 & 0 \\ 4 & -5 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 2 & 3 & 5 & 6 & 17 & 8 \\ -3 & 4 & 4 & 8 & 9 & 27 & 8 \\ 11 & -11 & -11 & -21 & -35 & -74 & -16 \\ -1 & 1 & 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

2.8 The `Mathematica` Program for Triangularization and Diagonalization

The triangularization and diagonalization of an integer matrix are obtained with elementary operations on the rows because with `Mathematica` it is more convenient to work on the rows of matrices instead of columns. Then, the corresponding column operations are obtained with appropriated transpositions.

The fundamental procedure is `annihilate`. The repeated applications of `annihilate` gives `leftUfactor`. These *primitive* procedures are applied to obtain triangularizations and diagonalizations of different forms.

Remark. In the following procedures the target is the application of elementary operations on an integer matrix `a` (input). The output is the unimodular matrix `e` that entails consideration of the elementary operations. Hence the output is not the rearranged matrix, say `b`. To get `b` one has to calculate the matrix product `b= e.a`.

2.8.1 *"annihilate"*

the procedure `annihilate` operates on two rows of a given integer matrix, say

```
r11  r12  r13  r14  r15  r16  ...   row r1
r21  r22  r23  r24  r25  r26  ...   row r2
```

where we suppose that r11 and r21 are different from zero. Then, with elementary operations it provides two rows with the structure

```
0    s12    s13   s14   s15  s16  ...    new row s1
@    s22    s23   s24   s25  s26  ...    new row s2
```

where @ is different from zero

Starting from the modified rows, `annihilate` gives the new rows

```
0    0    t13   t14   t15  t16  ...     new row t1
0    @    t23   t24   t25  t26  ...     new row t2
```

In a subsequent step, the rows are modified again to get

```
0    0    0   u14   u15  u16  ...     new row u1
0    0    @   u24   u25  u26  ...     new row u2
```

and so on.

Usage

```
e = annihilate[r1,r2,n1,n2,nr]
```

where

- `r1, r2` the rows,
- `n1, n2` are indexes of `r1, r2`,
- `nr` is the number of rows of the matrix,
- `e` is the unimodular matrix such that the matrix `b=e.a` provides the desired modification of the rows `r1, r2`.

Examples

Given the matrix

$$\mathbf{A} = \begin{array}{c} \left[\begin{array}{ccccccc} 0 & 5 & -8 & 2 & -1 & 3 & 1 \\ \mathbf{3} & \mathbf{1} & \mathbf{2} & -\mathbf{5} & \mathbf{7} & -\mathbf{4} & \mathbf{2} \\ 7 & 4 & -1 & -2 & 0 & 5 & 0 \\ 0 & 0 & 3 & 1 & 1 & 9 & 3 \\ \mathbf{8} & \mathbf{5} & -\mathbf{1} & \mathbf{6} & \mathbf{2} & \mathbf{1} & \mathbf{5} \\ 0 & 0 & 7 & 4 & 1 & 1 & 2 \end{array}\right] \end{array} \begin{array}{l} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \\ \mathbf{r}_4 \\ \mathbf{r}_5 \\ \mathbf{r}_6 \end{array}$$

and starting from the rows

$$\mathbf{r}_2 = [3 \quad 1 \quad 2 \quad -5 \quad 7 \quad -4 \quad 2]$$
$$\mathbf{r}_5 = [8 \quad 5 \quad -1 \quad 6 \quad 2 \quad 1 \quad 5]$$

`annihilate` provides a unimodular matrix $\mathbf{E}$ of dimensions 6×6, such that in the matrix

$$\mathbf{B} = \mathbf{E}\mathbf{A} \tag{2.4}$$

the corresponding rows become

$$\mathbf{s}_2 = [0 \quad 7 \quad -19 \quad 58 \quad -50 \quad 35 \quad -1]$$
$$\mathbf{s}_5 = [1 \quad -2 \quad 7 \quad -21 \quad 19 \quad -13 \quad 1]\,.$$

In this case the unimodular matrix $\mathbf{E}$ is given by the product of six unimodular matrices, namely

$$\mathbf{E} = \mathbf{E}_6\,\mathbf{E}_5\,\mathbf{E}_4\,\mathbf{E}_3\,\mathbf{E}_2\,\mathbf{E}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -8 & 0 & 0 & 3 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

corresponding to the following elementary operations:

1) permutation of the rows $\mathbf{r}_2$ and $\mathbf{r}_5$,
2) addition of row $\mathbf{r}_5$, multiplied by -2, to row $\mathbf{r}_2$,
3) permutation of the rows $\mathbf{r}_2$ and $\mathbf{r}_5$,
4) addition of row $\mathbf{r}_5$, multiplied by -1, to row $\mathbf{r}_2$,
5) permutation of the rows $\mathbf{r}_2$ and $\mathbf{r}_5$,
6) addition of row $\mathbf{r}_5$ multiplied by -2 to row $\mathbf{r}_2$.

For instance, 4), is obtained with the unimodular matrix

$$\mathbf{E}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}\,.$$

The global unimodular matrix $\mathbf{E}$ allows the desired conversion of the rows. In fact, relation (2.4) becomes explicitly

$$\mathbf{B} = \begin{bmatrix} 0 & 5 & -8 & 2 & -1 & 3 & 1 \\ \mathbf{0} & \mathbf{7} & \mathbf{-19} & \mathbf{58} & \mathbf{-50} & \mathbf{35} & \mathbf{-1} \\ 7 & 4 & -1 & -2 & 0 & 5 & 0 \\ 0 & 0 & 3 & 1 & 1 & 9 & 3 \\ \mathbf{1} & \mathbf{-2} & \mathbf{7} & \mathbf{-21} & \mathbf{19} & \mathbf{-13} & \mathbf{1} \\ 0 & 0 & 7 & 4 & 1 & 1 & 2 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -8 & 0 & 0 & 3 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 5 & -8 & 2 & -1 & 3 & 1 \\ \mathbf{3} & \mathbf{1} & \mathbf{2} & \mathbf{-5} & \mathbf{7} & \mathbf{-4} & \mathbf{2} \\ 7 & 4 & -1 & -2 & 0 & 5 & 0 \\ 0 & 0 & 3 & 1 & 1 & 9 & 3 \\ \mathbf{8} & \mathbf{5} & \mathbf{-1} & \mathbf{6} & \mathbf{2} & \mathbf{1} & \mathbf{5} \\ 0 & 0 & 7 & 4 & 1 & 1 & 2 \end{bmatrix}\,.$$

We continue the application of `annihilate` to the rows $(\mathbf{r}_4, \mathbf{r}_6)$ of the matrix $\mathbf{B}$

$$\mathbf{r}_4 = [0 \quad 0 \quad 3 \quad 1 \quad 1 \quad 9 \quad 3]$$
$$\mathbf{r}_6 = [0 \quad 0 \quad 7 \quad 4 \quad 1 \quad 1 \quad 2]$$

where the first two elements of both rows are zero. In this case “annihilate” starts from the third elements and provides a unimodular matrix $\tilde{\mathbf{E}}$ such that in the relation

$$\mathbf{C} = \tilde{\mathbf{E}}\mathbf{B} \tag{2.5}$$

the corresponding rows become

$$\mathbf{s}_4 = [0 \quad 0 \quad 0 \quad -5 \quad 4 \quad 60 \quad 15]$$
$$\mathbf{s}_6 = [0 \quad 0 \quad 1 \quad 2 \quad -1 \quad -17 \quad -4]$$

In fact, (2.5) gives explicitly

$$\begin{bmatrix} 0 & 5 & -8 & 2 & -1 & 3 & 1 \\ 3 & 1 & 2 & -5 & 7 & -4 & 2 \\ 7 & 4 & -1 & -2 & 0 & 5 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{3} & \mathbf{1} & \mathbf{1} & \mathbf{9} & \mathbf{3} \\ 8 & 5 & -1 & 6 & 2 & 1 & 5 \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{2} & \mathbf{-1} & \mathbf{-17} & \mathbf{-4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 0 & -3 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 5 & -8 & 2 & -1 & 3 & 1 \\ 3 & 1 & 2 & -5 & 7 & -4 & 2 \\ 7 & 4 & -1 & -2 & 0 & 5 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{3} & \mathbf{1} & \mathbf{1} & \mathbf{9} & \mathbf{3} \\ 8 & 5 & -1 & 6 & 2 & 1 & 5 \\ \mathbf{0} & \mathbf{0} & \mathbf{7} & \mathbf{4} & \mathbf{1} & \mathbf{1} & \mathbf{2} \end{bmatrix}$$

2.8.2 “leftUfactor” and companions

`leftUfactor` evaluates the unimodular matrix `e` that provides the upper–triangular form of a given matrix `a`, namely `u= e.a`, where `u` is upper triangular.

Usage

```
e = leftUfactor[a,adj]
```

where

- `a` is the given integer matrix,
- `adj` is an *optional* argument with default value `adj = 1`, which provides the canonical form, while with `adj = 0` **U** and **L** are not necessarily in canonical form,
- `e` is the unimodular matrix.

Hence, two versions are available

```
e = leftUfactor[a]
e = leftUfactor[a,1].
```

Illustration of the procedure

Given the matrix $\mathbf{A}$, the target is to get a unimodular matrix $\mathbf{E}$, such that matrix $\mathbf{U} = [u_{ij}] = \mathbf{E}\,\mathbf{A}$ where the elements u_{ij} with $j < i$ are zero. This is obtained using iteratively the procedure `annihilate`.

Example 2.13. Consider the square matrix:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 0 & 2 \\ 4 & -1 & 3 \end{bmatrix} .$$

Then, the following operations are needed:

1) application of "annihilate" to the rows $(\mathbf{r}_2, \mathbf{r}_1)$ of $\mathbf{A}$. This produces the unimodular $\mathbf{E}_1$.
2) evaluation of the matrix $\mathbf{A}_1 = \mathbf{E}_1\,\mathbf{A}$
3) application of "annihilate" to the rows $(\mathbf{r}_3, \mathbf{r}_1)$ of $\mathbf{A}_1$. This produces the unimodular $\mathbf{E}_2$.
4) evaluation of the matrix $\mathbf{A}_2 = \mathbf{E}_2\,\mathbf{A}_1$
5) application of "annihilate" to the rows $(\mathbf{r}_3, \mathbf{r}_2)$ of $\mathbf{A}_2$. This produces the unimodular $\mathbf{E}_3$.
6) evaluation of $\mathbf{A}_2$ as $\mathbf{A}_3 = \mathbf{E}_3\,\mathbf{A}_2$.

The global unimodular matrix $\mathbf{E}$ is

$$\mathbf{E} = \mathbf{E}_3\,\mathbf{E}_2\,\mathbf{E}_1 .$$

In the specific case one gets

$$\mathbf{E} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & -1 & 1 \\ -1 & 3 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 3 & -2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} .$$

In general, for a matrix $\mathbf{A}_{m\times n}$, which is not necessarily square, where the numbers of rows is m, "annihilate" must be applied $m(m-1)/2$ times, where m is the numbers of rows of $\mathbf{A}$. The option `adj = 1` is obtained with the subsequent application of `setcanonic`.

The procedure "setcanonic"

After the application of "leftUfactor" to get the upper-triangular form, `setcanonic` provides the canonical form. In the case of a nonsingular square matrix $\mathbf{A}_{m\times m}$ the following constraints are pursued for $\mathbf{U}$

$$u_{ii} > 0\,, \qquad 0 \le u_{ij} < u_{ii} \quad (j > i) \quad \text{with } u_{ij} \text{ coprime with respect to } u_{ii} .$$

If $\mathbf{A}$ is singular, with rank $k < m$, only the elements u_{ij} for $i, j \le k$ are adjusted.

Usage

```
e1 = setcanonic[e_,a_]
```

where

- `e.a` is the matrix to be set in canonical form
- `e1` is the unimodular matrix such that `u=e1.e.a` gives the canonical form.

Illustration

Consider the matrix

$$\mathbf{A} = \begin{bmatrix} -3 & 2 & -3 \\ 0 & 4 & -6 \\ -1 & 5 & 0 \end{bmatrix}.$$

We first apply $\mathbf{E}_0 = \text{leftUfactor}[\mathbf{A}]$, to get the upper–triangular triangular form $\mathbf{B}$, specifically

$$\mathbf{B} = \mathbf{E}_0\,\mathbf{A} = \begin{bmatrix} -1 & 5 & 0 \\ 0 & -1 & -21 \\ 0 & 0 & -90 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 3 & -3 \\ 4 & 13 & -12 \end{bmatrix} \begin{bmatrix} -3 & 2 & -3 \\ 0 & 4 & -6 \\ -1 & 5 & 0 \end{bmatrix}.$$

Then we apply “setcanonic”. The first operation performed by this macro is the evaluation of a unimodular matrix $\mathbf{E}_1$, which ensures the diagonal elements b_{ii} are positive: $\mathbf{B}_1 = \mathbf{E}_1\,\mathbf{B}$. In the specific case we have

$$\mathbf{B}_1 = \begin{bmatrix} 1 & -5 & 0 \\ 0 & 1 & 21 \\ 0 & 0 & 90 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} -1 & 5 & 0 \\ 0 & -1 & -21 \\ 0 & 0 & -90 \end{bmatrix}.$$

The second operation of “setcanonic” is to find another unimodular matrix that ensures that the elements b_{ij} are coprime with respect to b_{ii}, for $j > i$, starting from the element b_{12} of the matrix $\mathbf{B}_1$. This is achieved by adding the second row, multiplied by 5, to the first row. This is obtained with a unimodular matrix $\mathbf{V}_1$ which gives $\mathbf{B}_2 = \mathbf{V}_1\,\mathbf{B}_1$, and specifically

$$\mathbf{B}_2 = \begin{bmatrix} 1 & 0 & 105 \\ 0 & 1 & 21 \\ 0 & 0 & 90 \end{bmatrix} = \begin{bmatrix} 1 & 5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -5 & 0 \\ 0 & 1 & 21 \\ 0 & 0 & 90 \end{bmatrix}.$$

Then, we consider the element b_{13} of the matrix $\mathbf{B}_2$ and we realize that it is sufficient to subtract the third row from the first one. In terms of matrices we have $\mathbf{B}_3 = \mathbf{V}_2\,\mathbf{B}_2$ and specifically

$$\mathbf{B}_3 = \begin{bmatrix} 1 & 0 & 15 \\ 0 & 1 & 21 \\ 0 & 0 & 90 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 105 \\ 0 & 1 & 21 \\ 0 & 0 & 90 \end{bmatrix}.$$

The final result of "setcanonic" is the matrix $\mathbf{E} = \mathbf{V}_2\,\mathbf{V}_1\,\mathbf{E}_1$ and specifically

$$\mathbf{E} = \begin{bmatrix} -1 & -5 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

2.8.3 Procedures obtained from `leftUfactor`

From "leftUfactor" following procedures are obtained using the row inversion ("rowrev") and the columns inversions ("colrev"), together with the transposition operation, namely

```
rightLfactor[a_, adj_:0 ]:= Transpose[leftUfactor[
                            Transpose[a],adj]]

rowrev[a_]:=Reverse[a]

colrev[a_]:=Transpose[Reverse[Transpose[a]]]

leftLfactor[a_, adj_:0 ]:=Module[{},
            e=rowrev[leftUfactor[colrev[a],adj]];
    {m,n}=Dimensions[a];
    If[m>n,
       esup=Take[e,{1,m-n}];
               einf=Take[e,{m-n+1,m}];
               enew=Join[einf,esup];
       e=enew];
       e]
rightUfactor[a_, adj_:0 ]:=colrev[rightLfactor[
                                 rowrev[a],adj]]
```

Then `hermite`, `triangU` and `triangL` are obtained with the program

```
hermite[a_,form_,adj_:0]:=Module[{e},
   Which[form == 3,
       e = leftUfactor[a,adj]; {Inverse[e], e . a},
         form == 4,
       e = leftLfactor[a,adj]; {Inverse[e], e . a},
         form == 1,
       e = rightUfactor[a,adj]; { Inverse[e], a.e},
         form == 2,
```

```
    e = rightLfactor[a,adj]; {Inverse[e], a.e}
                      ] ]
  triangU[a_,adj_:1]:= a . rightUfactor[a,adj]
  triangL[a_,adj_:1]:= a . rightLfactor[a,adj]
```

2.8.4 Smith Diagonalization

In the procedure `smith[a,adj]` the matrix `diag` is initialized to `a` and then it is triangulated, lower and upper, using respectively `rightLfactor` and `leftUfactor` until it becomes diagonal; `diagcheck` is used to check the diagonal form. The corresponding column and row operations are reported in the unimodular matrices `e1` and `u1`. Finally, if `adj = 1` the canonical form is obtained by changing the signs of the negative elements of the diagonal; these changes are reported in `e1` if $m \geq n$ and in `u1` if $m < n$ ($m \times n$ are the dimension of `a`).

Example 2.14. Consider the 4×4 matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} 3 & 3 & 2 & 0 \\ 1 & 7 & 5 & 1 \\ 0 & 3 & 0 & 7 \\ 1 & 1 & 3 & 2 \end{bmatrix}$$

First the matrix $\mathbf{E}_2$ and the product $\mathbf{D}_1 = \mathbf{A}\,\mathbf{E}_2$ are evaluated

$$\mathbf{E}_2 = \text{rightLfactor}\,[\mathbf{A}] = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & -2 & 2 & 7 \\ -1 & 3 & -3 & -9 \\ 0 & 0 & 1 & -3 \end{bmatrix}$$

$$\mathbf{D}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -4 & 1 & 0 & 0 \\ 0 & -6 & 13 & 0 \\ -2 & 7 & -5 & -27 \end{bmatrix} = \begin{bmatrix} 3 & 3 & 2 & 0 \\ 1 & 7 & 5 & 1 \\ 0 & 3 & 0 & 7 \\ 1 & 1 & 3 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & -2 & 2 & 7 \\ -1 & 3 & -3 & -9 \\ 0 & 0 & 1 & -3 \end{bmatrix}$$

Analogously the matrices $\mathbf{F}_2$ and $\mathbf{D}_2 = \mathbf{F}_2\,\mathbf{D}_1$ are evaluated

$$\mathbf{F}_2 = \text{leftUfactor}\,[\mathbf{D}_1] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 \\ -82 & -23 & 2 & 5 \\ -218 & -61 & 5 & 13 \end{bmatrix}$$

$$\mathbf{D}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -\mathbf{135} \\ 0 & 0 & 0 & -351 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 \\ -82 & -23 & 2 & 5 \\ -218 & -61 & 5 & 13 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -4 & 1 & 0 & 0 \\ 0 & -6 & 13 & 0 \\ -2 & 7 & -5 & -27 \end{bmatrix} .$$

Then, “diagcheck” is applied to $\mathbf{D}_2$. The check is negative because in $\mathbf{D}_2$ the entry $(3,4)$ is $-\mathbf{135} \neq 0$. Then with `leftUfactor` $\mathbf{D}_3 = \mathbf{D}_2\,\mathbf{E}_3$ is evaluated

$$\mathbf{E}_3 = \text{rightLfactor}\,[\mathbf{D}_2] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 135 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{D}_3 = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & -\mathbf{351} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -135 \\ 0 & 0 & 0 & -351 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 135 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The next operation is the evaluation of $\mathbf{F}_3$ and $\mathbf{D}_4 = \mathbf{F}_3\,\mathbf{D}_3$; since $\mathbf{D}_3$ is already diagonal, one gets $\mathbf{F}_3 = \mathbf{I}_4$ and $\mathbf{D}_4 = \mathbf{D}_3$.

Finally, to get positive diagonal entries, the matrices $\mathbf{E}_0$ and $\boldsymbol{\Delta} = \mathbf{D}_4\,\mathbf{E}_0$ are evaluated

$$\mathbf{E}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$$\boldsymbol{\Delta} = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & \mathbf{351} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -351 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

The columns operations are reported in the matrix $\mathbf{E}_1 = \mathbf{E}_2\,\mathbf{E}_3\,\mathbf{E}_0$ and the row operation on the matrix $\mathbf{F}_1 = \mathbf{F}_3\,\mathbf{F}_2$, specifically

$$\mathbf{E}_1 = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & -2 & 2 & 7 \\ -1 & 3 & -3 & -9 \\ 0 & 0 & 1 & -3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 135 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & -2 & 2 & -277 \\ -1 & 3 & -3 & 414 \\ 0 & 0 & 1 & -132 \end{bmatrix}.$$

$$\mathbf{F}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 \\ -82 & -23 & 2 & 5 \\ -218 & -61 & 5 & 13 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 \\ -82 & -23 & 2 & 5 \\ -218 & -61 & 5 & 13 \end{bmatrix}$$

The decomposition so obtained is $\mathbf{F}_1\,\mathbf{A}\,\mathbf{E}_1 = \boldsymbol{\Delta}$, specifically

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 \\ -82 & -23 & 2 & 5 \\ -218 & -61 & 5 & 13 \end{bmatrix}}_{\mathbf{F}_1} \underbrace{\begin{bmatrix} 3 & 3 & 2 & 0 \\ 1 & 7 & 5 & 1 \\ 0 & 3 & 0 & 7 \\ 1 & 1 & 3 & 2 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & -2 & 2 & -277 \\ -1 & 3 & -3 & 414 \\ 0 & 0 & 1 & -132 \end{bmatrix}}_{\mathbf{E}_1} = \underbrace{\begin{bmatrix} \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & \mathbf{351} \end{bmatrix}}_{\boldsymbol{\Delta}}.$$

To get the Smith decomposition one has to evaluate the inverse matrices $\mathbf{F} = \mathbf{F}_1^{-1}$ and $\mathbf{E} = \mathbf{E}_1^{-1}$ to get

$$\mathbf{A} = \mathbf{F}\boldsymbol{\Delta}\mathbf{E}$$

$$\begin{bmatrix} 3 & 3 & 2 & 0 \\ 1 & 7 & 5 & 1 \\ 0 & 3 & 0 & 7 \\ 1 & 1 & 3 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -4 & 1 & 0 & 0 \\ 0 & -6 & 13 & -5 \\ -2 & 7 & -5 & 2 \end{bmatrix} \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & \mathbf{351} \end{bmatrix} \begin{bmatrix} 3 & 3 & 2 & 0 \\ 13 & 19 & 13 & 1 \\ -264 & -396 & -264 & 1 \\ -2 & -3 & -2 & 0 \end{bmatrix}.$$

Chapter 3
Generation of Lattice Points

Given a lattice G of $\mathbb{R}^m$, specified by a basis $\mathbf{G}$, and given a subset (region) $\mathcal{R}$ of $\mathbb{R}^m$, we want to generate the points of G belonging to $\mathcal{R}$, that is, the set $\mathcal{P} = G \cap \mathcal{R}$. The solution of this problem is not trivial. The evaluation of single points according to

$$\mathbf{t} = \mathbf{G}\,\mathbf{h}\,, \qquad \mathbf{h} \in \mathbb{Z}^m$$

is immediate, but the problem is the limitation of the m–tuples of integers $\mathbf{h}$ to limit the generation to the desired region. In the simplest case the region $\mathcal{R}$ is a parallelepiped and the generation is obtained starting from the lower triangular form of the basis.

The chapter deals with the implementation in `Mathematica` of the generation of the points a full dimensional lattice, but also of a reduced-dimensional lattice, where the problem is somewhat complicated.

Procedures of this chapter

- `latticepoints2D`: gives the points of a 2D lattice limited to a rectangular region (Sect. 3.2);
- `reglatticepoints`: gives the points of an mD lattice limited to a rectangular region (Sect. 3.3);
- `degdecomposition`: decompose an $m \times n$ integer matrix with rank k containing a $k \times k$ minor (Sect. 3.4);
- `deglatticepoints`: gives the points of a reduced dimensional lattice limited to a rectangular region (Sect. 3.4);
- `latticepoints`: unifies the previous procedures (Sect. 3.5);
- `writelatticepoints`: writes (in TEX) the lattice points (Sect. 3.7);
- `showlatticepoints`: gives the graphic representation of the lattice points in the cases 2D and 3D (Sect. 3.7).

3.1 Formulation of the Problem

Is is convenient to recall that an mD lattice is generated according to

$$G = \{\mathbf{G}\,\mathbf{h} \,|\, \mathbf{h} \in \mathbb{Z}^m\} = \mathbf{G}\,\mathbb{Z}^m \ . \tag{3.1}$$

where $\mathbf{G}$ is a basis. This relation gives *all* the points of the lattice. Every basis of G generates all the points of G.

The general problem is: given a region $\mathcal{R}$ and a property Π, find the subset

$$\mathcal{P} = \{G \cap \mathcal{R} | \Pi\} \ .$$

For instance, if $\mathcal{R}$ parallelepiped and $\Pi = \{\text{primitive points}\}$, then $\mathcal{P}$ consists of the *primitive* points of G that belong to $\mathcal{R}$.

The solution is obtained using a convenient basis $\mathbf{G}_0$ (lower triangular) and a "simple" region (parallelepiped) $\mathcal{R}_0 \supset \mathcal{R}$. Then, the points are obtained by eliminating from $\mathcal{R}_0$ the points that do not have the desired property.

The convenient region $\mathcal{R}_0$ is usually given by an orthogonal parallelepiped

$$\mathcal{R}_0 = [t_{1_{min}}, t_{1_{max}}] \times \ldots \times [t_{m_{min}}, t_{m_{max}}] \ . \tag{3.2}$$

3.1.1 Primitive points

From Chapter 16 of the *Unified Signal Theory* we recall:

Definition 3.1. A point of a lattice G is *primitive* if the segment connecting the point to the origin does not contain other lattice points, origin excepted.

Fig.3.1 shows the primitive points of the 2D lattice $\mathbb{Z}_4^3(d_1, d_2)$.

Proposition 3.1. *A point* $\mathbf{t} = \mathbf{Gh}$ *of a lattice* $G = \mathbf{G}\mathbb{Z}^m$ *is primitive if* $\mathsf{GCD}(\mathbf{h}) = 1$, *where* $\mathsf{GCD}(\mathbf{h})$ *is the greatest common divisor of the integers* $\mathbf{h} = (h_1, \ldots, h_m)$.

It is convenient to precise what the GCD means when some of the h_i of $\mathbf{h}$ are zero. For instance, if $m = 3$ and $(h_1, h_2, h_3) = (6, 0, 3) = 3\,(2, 0, 1)$ it does not give a primitive point, but $(h_1, h_2, h_3) = (8, 0, 5)$ does give a primitive point. In fact, we find with `Mathematica`)

$$\mathrm{GCD}(6, 0, 3) = 3 \quad \text{and} \quad \mathrm{GCD}(8, 0, 5) = 1$$

while $\mathrm{GCD}(0, 0, 0) = 0$.

3.2 Points of 2D and 3D Lattices

3.2.1 Role of the basis in point generation

Let $\mathbf{G}$ be a 2×2 real matrix

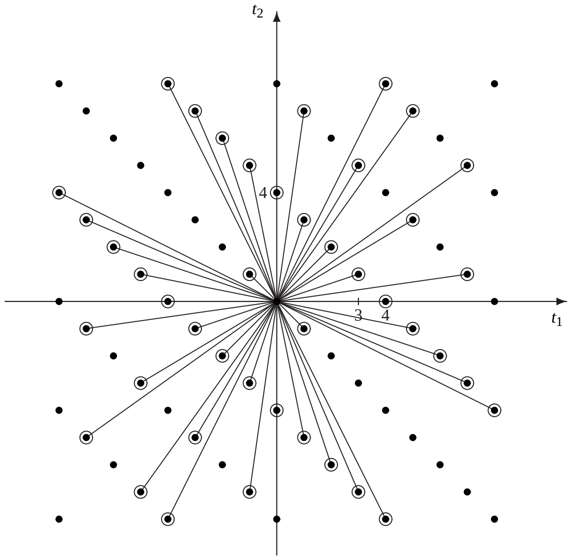


Fig. 3.1 Primitive points of the lattice $\mathbb{Z}_4^3(d_1,d_2)$

• $\in J = \mathbb{Z}_1^3(1,1)$
◉ primitive points

$$\mathbf{G} = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}$$

which is assumed as regular, that is, with $\mathrm{d}(G) = |g_{11}g_{22} - g_{21}g_{12}| > 0$. The corresponding lattice has the points with coordinates

$$\begin{aligned} t_1 &= g_{11}h_1 + g_{12}h_2\,, \qquad h_1, h_2 \in \mathbb{Z} \\ t_2 &= g_{21}h_1 + g_{22}h_2 \end{aligned} \tag{3.3}$$

If in (3.3) (h_1,h_2) is limited to a rectangular region according to

$$h_{1_{min}} \le h_1 \le h_{1_{max}}\,, \qquad h_{2_{min}} \le h_2 \le h_{2_{max}}$$

the corresponding region in G is not rectangular, in general. For instance, with the basis

$$\mathbf{G} = \begin{bmatrix} 4 & 2 \\ 2 & 6 \end{bmatrix} \quad , \qquad d(G) = 20 \tag{3.4}$$

the limitation of (h_1,h_2) according to

$$-4 \le h_1 \le 4 \qquad -4 \le h_2 \le 4 \tag{3.5}$$

produces the region shown in Fig. 3.2a, which is the parallelogram determined by the basis (3.4). If we change the basis, e.g.

$$\mathbf{G}_L = \begin{bmatrix} 4 & 6 \\ 2 & 8 \end{bmatrix} \tag{3.6}$$

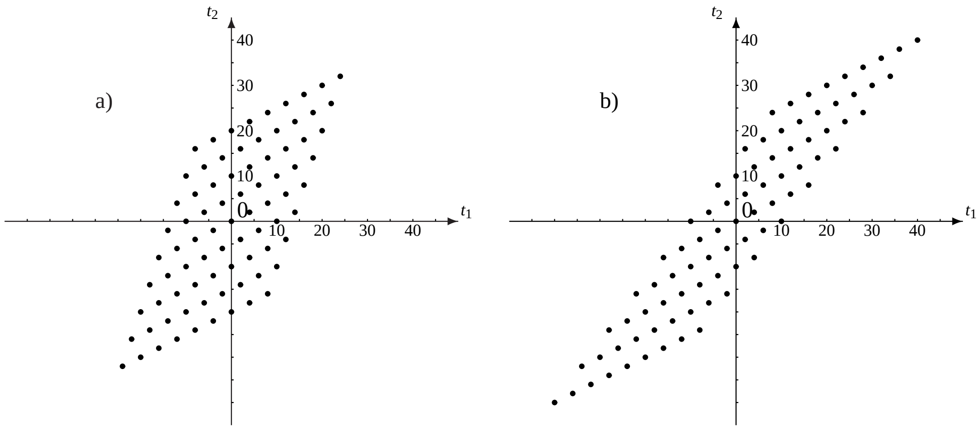


Fig. 3.2 Effect of basis change in a 2D lattice

the limitation (3.5), produces the points shown in Fig. 3.2b, which again are not contained in a rectangular region.

To get a rectangular region, we use a lower-triangular canonical basis, say

$$\mathbf{G}_L = \begin{bmatrix} b_{11} & 0 \\ b_{21} & b_{22} \end{bmatrix}$$

where

$$b_{11} > 0\,, \qquad b_{22} > 0\,, \qquad 0 \le b_{21} < b_{22}\;.$$

Let

$$\mathcal{R} = [t_{1_{min}}, t_{1_{max}}] \times [t_{2_{min}}, t_{2_{max}}]$$

the desired rectangular region. The point coordinates are given by

$$\begin{cases} t_1 = b_{11}h_1 \\ t_2 = b_{21}h_1 + b_{22}h_2 \end{cases} \qquad h_1, h_2 \in \mathbb{Z}\;. \tag{3.7}$$

From the first relation, we evaluate $h_{1_{min}}$ and $h_{1_{max}}$ corresponding to $t_{1_{min}}$ and $t_{1_{max}}$, namely

$$h_{1_{min}} = t_{1_{min}}/b_{11} \qquad h_{1_{max}} = t_{1_{max}}/b_{11}\;. \tag{3.8a}$$

Then, as h_1 varies between $h_{1_{min}}$ and $h_{1_{max}}$, one evaluates, for each h_1 the value of $h_{2_{min}}$ and $h_{2_{max}}$ from the second equation, that is,

$$h_{2_{min}} = (t_{2_{min}} - b_{21}h_1)/b_{22} \qquad h_{2_{max}} = (t_{2_{max}} - b_{21}h_1)/b_{22} \quad . \tag{3.8b}$$

Now, $h_{1_{min}}$, $h_{1_{max}}$, $h_{2_{min}}$ and $h_{2_{max}}$ should be integers. But, e.g., $h_{1_{min}} = t_{1_{min}}/b_{11}$ is not an integer, in general, and we have to impose that $h_{1_{min}}$ be *the smallest integer* $\ge t_{1_{min}}/b_{11}$.

The correct formulation is obtained with the function `Ceiling` and `Floor` of `Mathematica`, and (3.8) are reformulated as follows

$$\begin{aligned} h_{1_{min}} &= \text{Ceiling}[t_{1_{min}}/b_{11}] \\ h_{1_{max}} &= \text{Floor}[t_{1_{max}}/b_{11}] \\ h_{2_{min}} &= \text{Ceiling}[(t_{2_{min}} - b_{21}h_1)/b_{22}] \\ h_{2_{max}} &= \text{Floor}[(t_{2_{max}} - b_{21}h_1)/b_{22}] \end{aligned} \tag{3.9}$$

Reconsidering the previous example, with the lower-triangular basis (3.6), relations (3.7) become

$$\begin{cases} t_1 = 2h_1 \\ t_2 = 6h_1 + 10h_2 \end{cases}$$

Considering the region

$$\mathcal{R} = [t_{1_{min}}, t_{1_{max}}] \times [t_{2_{min}}, t_{2_{max}}] = [-9, 9] \times [-9, 9]$$

(3.9) give

$$h_{1_{min}} = \text{Ceiling}[\tfrac{-9}{2}] = -4 \qquad h_{1_{max}} = \text{Floor}[\tfrac{9}{2}] = 4\ .$$

$$h_{2_{min}} = \text{Ceiling}\left[\frac{-9-6\cdot(-4)}{10}\right] = \text{Ceiling}\left[\frac{15}{10}\right] = 2$$

$$h_{2_{max}} = \text{Floor}\left[\frac{9-6\cdot(-4)}{10}\right] = \text{Floor}\left[\frac{33}{10}\right] = 3$$

Note that $h_1 = -4$ give two values of h_2, which form the pairs $(-4, 2)$, $(-4, 3)$, corresponding to the points $\mathbf{t}$: $(-8, -4)$, $(-8, 6)$, as shown Fig.3.3.

The results are summarized by the following table:

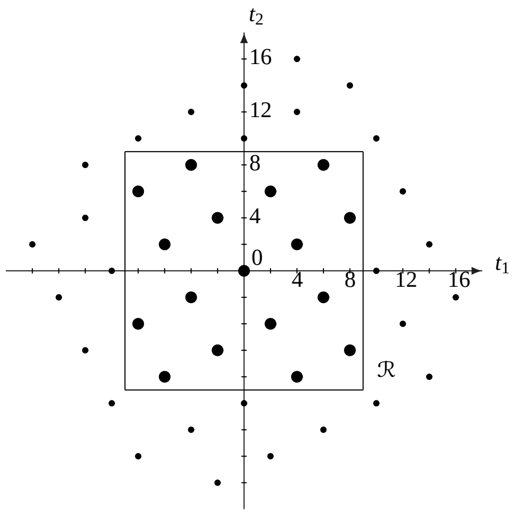


Fig. 3.3 2D lattice and its points in the rectsngular region $\mathcal{R}$

h_1	h_{2min}	h_{2max}	**h**	**t**
−4	2	3	(−4,2) (−4,3)	(−8,4) (−8,6)
−3	1	2	(−3,1) (−3,2)	(−6,−8) (−6,2)
−2	1	2	(−2,1) (−2,2)	(−4,−2) (−4,8)
−1	0	1	(−1,0) (−1,1)	(−2,−6) (−2,4)
0	0	0	(0,0)	(0,0)
1	−1	0	(1,−1) (1,0)	(2,−4) (2,6)
2	−2	−1	(2,−2) (2,−1)	(4,−8) (4,2)
3	−2	−1	(3,−2) (3,−1)	(6,−2) (6,8)
4	−3	−2	(4,−3) (4,−2)	(8,−6) (8,4)

3.2.2 The Mathematica procedure “latticepoints2D”

Usage

```
p = latticepoints2D[a,region,op]
```

where

- `a` is a basis of the lattice,
- `region` is the rectangular region specified in the form: `{xmin,xmax,ymin,ymax}`
- `op` is an optional argument with default value `op = 0`; with `op = 1` it gives only the *primitive* points,
- `p` gives the lattice points in the format `List` of `Mathematica`.

Hence one gets the three possible versions:

`p = latticepoints2D[a,region]`,

`p = latticepoints2D[a,region,1]`,

`p = latticepoints2D[a,region,2]`.

Method. First the lower-triangular form is obtained. Then, the lattice points are generated according (3.7) and (3.9). The procedure is therefore:

```
latticepoints2D[a_,region_,op_:0]:=
        Module[{t1min,t1max,t2min,t2max,b, list,
```

```
        h1min,h1max,h2min,h2max},
{t1min,t1max,t2min,t2max} = region;
b = triangL[a];
list = {};
h1min =Ceiling[ t1min/b[[1,1]]]; h1max = t1max/b[[1,1]];
Do[ h2min = Ceiling[ (t2min-b[[2,1]] h1 )/b[[2,2]]] ;
    h2max = (t2max-b[[2,1]] h1)/b[[2,2]];
      Do[ If[op==0 || GCD[h1,h2] == 1,
             list = Append[list, b . {h1,h2}] ],
        {h2,h2min,h2max} ],
  {h1,h1min,h1max} ];
list]
```

3.2.3 Demo of "latticepoints2D"

To get the demo `writelatticepoints` the following is needed in `Mathematica`

- 1) define the basis (integer square matrix) `a`,
- 2) choose the region `region`,
- 3) choose the optional argument `op`,
- 4) write the statement `writelatticepoints2D[a,region,op]`.

Example 3.1. The first example of demo consider a 2D lattice. The program for `Mathematica` is

```
Get["demo2macro.m"]
lab="LP204"
initdemo2
p22 = {{2, 1},
       {1, 3}}
region={0,15,0,15}
writelatticepoints2D[p22,region,2]b
completedemo2
```

Hence TEX gives

Lattice points: run $\boxed{\textbf{LP204}}$ (May 26, 2011)
Given an integer matrix (basis), which identify the lattices G and a rectangular region $\mathbf{r}$:

$$\mathbf{G} = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \qquad \mathbf{r} = \begin{bmatrix} 0 & 15 & 0 & 15 \end{bmatrix}$$

The procedure `writelatticepoints` gives the lattice points generated by $\mathbf{G}$ in the region (extrema included).

$op = 2$: all lattice points with primitive lattice points marked
number of points: 52

$$\begin{bmatrix} 0 & \mathbf{0} & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{2} & 2 & \mathbf{2} & \mathbf{3} & 3 & \mathbf{3} & 4 & \mathbf{4} & 4 & \mathbf{5} & \mathbf{5} & \mathbf{5} & 5 \\ 0 & \mathbf{5} & 10 & 15 & \mathbf{3} & \mathbf{8} & \mathbf{13} & \mathbf{1} & 6 & \mathbf{11} & \mathbf{4} & 9 & \mathbf{14} & 2 & \mathbf{7} & 12 & \mathbf{0} & \mathbf{5} & \mathbf{10} & 15 \end{bmatrix}$$

$$\begin{bmatrix} 6 & 6 & \mathbf{6} & \mathbf{7} & \mathbf{7} & \mathbf{7} & 8 & \mathbf{8} & 8 & \mathbf{9} & \mathbf{9} & 9 & 10 & 10 & 10 & \mathbf{10} & \mathbf{11} & \mathbf{11} & \mathbf{11} & \mathbf{12} \\ 3 & 8 & \mathbf{13} & \mathbf{1} & \mathbf{6} & \mathbf{11} & 4 & \mathbf{9} & 14 & \mathbf{2} & \mathbf{7} & 12 & 0 & 5 & 10 & \mathbf{15} & \mathbf{3} & \mathbf{8} & \mathbf{13} & \mathbf{1} \end{bmatrix}$$

$$\begin{bmatrix} 12 & \mathbf{12} & \mathbf{13} & \mathbf{13} & \mathbf{13} & 14 & 14 & 14 & 15 & \mathbf{15} & \mathbf{15} & 15 \\ 6 & \mathbf{11} & \mathbf{4} & \mathbf{9} & \mathbf{14} & 2 & 7 & 12 & 0 & \mathbf{5} & \mathbf{10} & 15 \end{bmatrix}$$

In this case `op=2` and therefore all the lattice points are given, with the primitive points in boldface.

Example 3.2. In previous example, if in place of `writelatticepoints` we write `showlatticepoints` we obtain also the graphic representation of the lattice points.

Lattice points: run $\boxed{\textbf{LP206}}$ (May 26, 2011)
Given an integer matrix (basis), which identify the lattices G and a rectangular region $\mathbf{r}$:

$$\mathbf{G} = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \qquad \mathbf{r} = \begin{bmatrix} 0 & 15 & 0 & 15 \end{bmatrix}$$

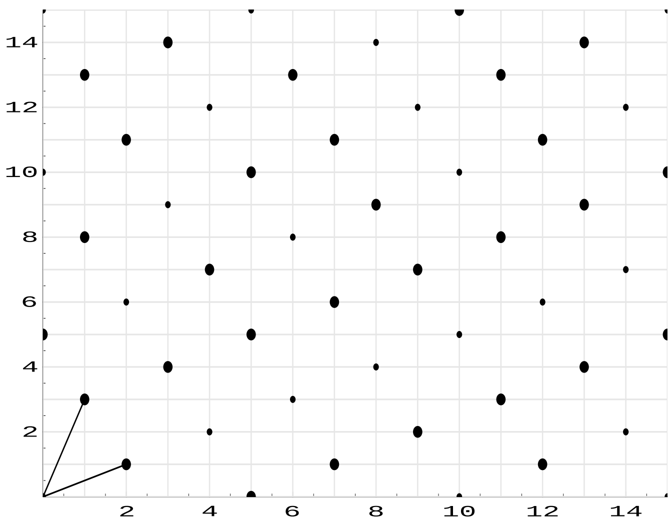
The procedure `writelatticepoints` gives the lattice points generated by $\mathbf{G}$ in the region (extrema included).

$op = 2$: all lattice points with primitive lattice points marked
number of points: 52

$$\begin{bmatrix} 0 & \mathbf{0} & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{2} & 2 & \mathbf{2} & \mathbf{3} & 3 & \mathbf{3} & 4 & \mathbf{4} & 4 & \mathbf{5} & \mathbf{5} & \mathbf{5} & 5 \\ 0 & \mathbf{5} & 10 & 15 & \mathbf{3} & \mathbf{8} & \mathbf{13} & \mathbf{1} & 6 & \mathbf{11} & \mathbf{4} & 9 & \mathbf{14} & 2 & \mathbf{7} & 12 & \mathbf{0} & \mathbf{5} & \mathbf{10} & 15 \end{bmatrix}$$

$$\begin{bmatrix} 6 & 6 & \mathbf{6} & \mathbf{7} & \mathbf{7} & \mathbf{7} & 8 & \mathbf{8} & 8 & \mathbf{9} & \mathbf{9} & 9 & 10 & 10 & 10 & \mathbf{10} & \mathbf{11} & \mathbf{11} & \mathbf{11} & \mathbf{12} \\ 3 & 8 & \mathbf{13} & \mathbf{1} & \mathbf{6} & \mathbf{11} & 4 & \mathbf{9} & 14 & \mathbf{2} & \mathbf{7} & 12 & 0 & 5 & 10 & \mathbf{15} & \mathbf{3} & \mathbf{8} & \mathbf{13} & \mathbf{1} \end{bmatrix}$$

$$\begin{bmatrix} 12 & \mathbf{12} & \mathbf{13} & \mathbf{13} & \mathbf{13} & 14 & 14 & 14 & 15 & \mathbf{15} & \mathbf{15} & 15 \\ 6 & \mathbf{11} & \mathbf{4} & \mathbf{9} & \mathbf{14} & 2 & 7 & 12 & 0 & \mathbf{5} & \mathbf{10} & 15 \end{bmatrix}$$



3.2.4 Points of a 3D lattice

We evaluate the canonical lower-triangular form, which allow expressing the coordinate in the form

$$\begin{cases} t_1 = b_{11}h_1 \\ t_2 = b_{21}h_1 + b_{22}h_2 \\ t_3 = b_{31}h_1 + b_{32}h_2 + b_{33}h_3 \end{cases} \qquad h_1, h_2, h_3 \in \mathbb{Z} \tag{3.10}$$

where $b_{ii} > 0$.

To get a rectangular region, the conditions on h_1, h_2, h_3 becomes

$$\begin{aligned} h1_{min} &= \text{Ceiling}[t1_{min}/b_{11}] \\ h2_{min} &= \text{Ceiling}[(t2_{min} - b_{21}h_1)/b_{22}] \\ h3_{min} &= \text{Ceiling}[(t3_{min} - b_{31}h_1 - b_{32}h_2)/b_{33}] \end{aligned}$$

$$\begin{aligned} h1_{max} &= \text{Floor}[t1_{max}/b_{11}] \\ h2_{max} &= \text{Floor}[(t2_{max} - b_{21}h_1)/b_{22}] \\ h3_{min} &= \text{Ceiling}[(t3_{min} - b_{31}h_1 - b_{32}h_2)/b_{33}] \ . \end{aligned}$$

Then, the implementation of `latticepoints3D` is obtained with three `Do[ ]` cycles, where the outer generates h_1 with fixed limits, the second generates h_2 with limits depending on h_1 and the inner generates h_3 with limits depending on h_1 and h_2.

Example 3.3. Let

$$\mathbf{G} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 5 & 2 \\ 3 & 2 & 1 \end{bmatrix} \qquad \mathcal{R} = [0,3] \times [0,3] \times [0,3] \; .$$

The lower-triangular form is

$$\mathbf{G}_L = \text{triangL} \; [\mathbf{G}] = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 3 & 0 \\ 1 & 1 & 2 \end{bmatrix} \; .$$

Hence the points $\mathbf{t} = (t_1, t_2, t_3)$ of the lattice are evaluate according to

$$\begin{cases} t_1 = h_1 \\ t_2 = 2h_1 + 3h_2 \\ t_3 = h_1 + h_2 + 2h_3 \end{cases} \; .$$

From the first one gets

$$h_{1_{min}} = 0 \qquad h_{1_{max}} = 3 \; .$$

Then the ranges of the h_1 and h_3 are evaluated according to the following table

h_1	h_2	$h_{3_{min}}$	$h_{3_{max}}$	$\mathbf{h}$	$\mathbf{t}$	GCD($\mathbf{h}$)
0	0	0	1	(0,0,0) (0,0,1)	(0,0,0) (0,0,2)	1 1
0	1	0	1	(0,1,0) (0,1,1)	(0,3,1) (0,3,3)	1 1
1	0	0	1	(1,0,0) (1,0,1)	(1,2,1) (1,2,3)	1 1
2	-1	0	1	(2,-1,0) (2,-1,1)	(2,1,1) (2,1,3)	1 1
3	-2	0	1	(3,-2,0) (3,-2,1)	(3,0,1) (3,0,3)	1 1
3	-1	-1	0	(3,-1,-1) (3,-1,0)	(3,3,0) (3,3,2)	1 1

3.3 Points of an mD Lattice

3.3.1 Generation from the lower-triangular basis

For a full-dimensional *m*D lattice we generalize the procedure seen in the cases 2D and 3D. Starting from a nonsingular basis $\mathbf{G}$ of dimensions $m \times m$, we find the lower-triangular form, that is,

$$\mathbf{G}_L = \begin{bmatrix} b_{11} & 0 & \cdots & 0 \\ b_{21} & b_{22} & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mm} \end{bmatrix}$$

where *the diagonal entries are positive*. Then the coordinates $\mathbf{t} = (t_1, \ldots, t_m)$ can be written in the form (which generalizes (3.7) and (3.10))

$$\begin{cases} t_1 = b_{11} h_1 \\ t_r = s_r + b_{rr} h_r \end{cases} \tag{3.11}$$

where

$$r = 2, \ldots, m \quad \text{e} \quad s_r = \sum_{k=1}^{r-1} b_{rk} h_k \, . \tag{3.11a}$$

Considering the rectangular region

$$\mathcal{R}_0 = [t_{1_{min}}, t_{1_{max}}] \times \ldots \times [t_{m_{min}}, t_{m_{max}}]$$

and considering that $b_{rr} > 0$, the ranges of the integers h_k are determined by

$$\begin{aligned} h_{1_{min}} &= \text{Ceiling}\,[t_{1_{min}}/b_{11}] \\ h_{1_{max}} &= \text{Floor}\,[t_{1_{max}}/b_{11}] \\ h_{r_{min}} &= \text{Ceiling}\,[(t_{r_{min}} - s_r)/b_{rr}] \\ h_{r_{max}} &= \text{Floor}\,[(t_{r_{max}} - s_r)/b_{rr}] \, . \end{aligned} \tag{3.12}$$

The limits $h_{r_{min}}$ and $h_{r_{max}}$ have the following dependence

- $h_{1_{min}}$ ed $h_{1_{max}}$ are fixed,
- $h_{2_{min}}$ ed $h_{2_{max}}$ depend on h_1
- $h_{3_{min}}$ ed $h_{3_{max}}$ depend on h_1, h_2
- in general $h_{r_{min}}$ and $h_{r_{max}}$ depend onh_1, h_2, $\ldots$, h_{r-1}.

Hence,h_1 will be generated in the outer cycle, h_2 in the second cycle, and so on until h_r, which will be generated in the inner cycle. The difficulty in the implementation is that the number of cycles is variable (in dependence of the dimension m. It would be possible to fix a maximum of m, say $m = 10$, and implement 10 distinct procedures, but this is not elegant.

Instead, we have composed a procedure ("writegattipoints"), which write the procedure for the specific m ("gattipoints") in a file. Then we invoke the procedure from the file. In conclusion, we have the structure

```
reglatticepoints[a_,region_,op_:0]:= Module[{m},
            m=Length[a];
            writegattipoints[m,op];
            Get["gatt.m"];
            gattipoints[a,region]
            ]
```

where "writegattipoints" write in "gatt.m" the procedure "gattipoints", specific for the given m.

3.3.2 The Mathematica procedure "reglatticepoints"

Usage

```
p = reglatticepoints[a,region,op]
```

where

- `a` is a basis of the lattice,
- `region` is the rectangular region specified in the form: `{rmin1,rmax1,rmin2,rmax2,...,rminm,rmaxm}`
- `op` is an optional argument with default value `op = 0`;
  with `op = 0` it gives all the points in the format `{t1,t2,...tm}`,
  with `op = 1` it gives only the *primitive* in the format `{t1,t2,...tm}`,
  with `op = 2` it gives all the points in the format `{{t1,t2,...tm},z}`,
  where z=0 indicates a non primitive point and z=1 indicates a primitive point.

For instance, with the basis and region

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix} \qquad \mathcal{R} = [0,5] \times [0,5]$$

With `op` omitted, we get all the points

```
p={{0, 0}, {0, 3}, {1, 0}, {1, 3}, {2, 0}, {2, 3}, {3, 0},
{3, 3}, {4, 0}, {4, 3}, {5, 0}, {5, 3}}
```

With `op=1`, we get only the primitive points

```
{{0, 3}, {1, 0}, {1, 3}, {2, 3}, {3, 3}, {4, 3}, {5, 3}}
```

With `op=2` we get all the points, made with the primitive points

```
{{{0, 0}, 0}, {{0, 3}, 1}, {{1, 0}, 1}, {{1, 3}, 1},
{{2, 0}, 0}, {{2, 3}, 1}, {{3, 0}, 0}, {{3, 3}, 1},
{{4, 0}, 0}, {{4, 3}, 1}, {{5, 0}, 0}, {{5, 3}, 1}}
```

In this case ($m = 2$) the procedure written in `gatti.m` is

```
gattipoints[a_,region_]:= Module[
   {t1min,t1max,t2min,t2max,t3min,t3max,
    b, list,
    h1min,h1max,h2min,h2max,h3min,h3max},
    b = triangL[a];
    list = {};
    h1min =Ceiling[ region[[1]]/b[[1,1]]];
     h1max = region[[2]]/b[[1,1]];
     Do[ h2min=Ceiling[(region[[3]]-b[[2,1]] h1)/b[[2,2]]];
         h2max=(region[[4]]-b[[2,1]] h1)/b[[2,2]];
     Do[ If[ GCD[h1,h2] == 1,
         list = Append[list, b . {h1,h2}] ],
        {h2,h2min,h2max,1} ],
        {h1,h1min,h1max,1} ];
list]
```

With the 5×5 matrix

$$\mathbf{B} = \begin{bmatrix} 2 & 0 & 0 & 1 & 5 \\ 0 & 3 & 0 & 2 & 7 \\ 0 & 0 & 2 & 3 & 11 \\ 0 & 1 & 2 & 3 & 13 \\ 1 & 3 & 2 & 0 & 3 \end{bmatrix} \tag{3.13}$$

the procedure written in `gatti.m` is

```
gattipoints[a_,region_]:= Module[
   {t1min,t1max,t2min,t2max,t3min,t3max,
    b, list,
    h1min,h1max,h2min,h2max,h3min,h3max},
    b = triangL[a];
    list = {};
    h1min =Ceiling[ region[[1]]/b[[1,1]]];
     h1max = region[[2]]/b[[1,1]];
     Do[ h2min=Ceiling[(region[[3]]-b[[2,1]] h1)/b[[2,2]]];
         h2max=(region[[4]]-b[[2,1]] h1)/b[[2,2]];
     Do[ h3min=Ceiling[(region[[5]]-b[[3,1]] h1-b[[3,2]]
               h2)/b[[3,3]]];
         h3max=(region[[6]]-b[[3,1]] h1-b[[3,2]] h2)/b[[3,3]];
     Do[ h4min=Ceiling[(region[[7]]-b[[4,1]]
                       h1-b[[4,2]] h2-b[[4,3]] h3)/b[[4,4]]];
         h4max=(region[[8]]-b[[4,1]] h1-b[[4,2]] h2-b[[4,3]]
                        h3)/b[[4,4]];
     Do[ h5min=Ceiling[(region[[9]]-b[[5,1]] h1-b[[5,2]]
         h2-b[[5,3]] h3-b[[5,4]] h4)/b[[5,5]]];
```

```
          h5max=(region[[10]]-b[[5,1]] h1-b[[5,2]]
  h2-b[[5,3]] h3-b[[5,4]] h4)/b[[5,5]];
      Do[ list = Append[list, b . {h1,h2,h3,h4,h5}],
         {h5,h5min,h5max,1} ],
         {h4,h4min,h4max,1} ],
         {h3,h3min,h3max,1} ],
         {h2,h2min,h2max,1} ],
         {h1,h1min,h1max,1} ];
 list]
```

3.4 Points of a Degenerate Lattice

In the *Unified Signal Theory* the basis is always assumed as regular and a degenerate lattice (reduced-dimensional lattice) is obtained acting on the signature. For instance, consider the matrix (3.13). With the signature is $H = \mathbb{Z}^5$, we get a full-dimensional lattice in $\mathbb{R}^5$, but we the signature $H = \mathbb{Z}^3 \times \mathbb{O}^2$, where $\mathbb{O} = \{0\}$, we get a 3D lattice in $\mathbb{R}^5$. An alternative way to specify a reduced-dimensional lattice is by a singular matrix with a rank $k < m$ and leaving the full signature $\mathbb{Z}^m$. In the example considered, a 3D lattice in $\mathbb{R}^5$ is obtained by a 5×5 matrix with rank $k = 3$ is needed.

In the implementation with `Mathematica` we follow this second possibility and more specifically we generate kD lattice of $\mathbb{R}^m$ from a basis $\mathbf{G}$ of dimension $m \times n$ and rank k, where e dove $k \le n \le m$.

This generation is based on a generalization of the Hermite triangular decomposition of the basis $\mathbf{G}$.

3.4.1 *Decomposition of a reduced-rank integer matrix*

Theorem 3.1. *An integer matrix* $\mathbf{G}$ *of dimensions* $m \times n$ *(*$n \le m$*) and rank* $k \le n$ *can be decomposed in the form*

$$\underset{m\times n}{\mathbf{G}} = \underset{m\times m}{\mathbf{P}} \begin{bmatrix} \underset{k\times k}{\mathbf{A}} & \underset{k\times n-k}{\mathbf{0}} \\ \underset{m-k\times k}{\mathbf{B}} & \underset{m-k\times n-k}{\mathbf{0}} \end{bmatrix} \underset{n\times n}{\mathbf{E}} = \underset{m\times m}{\mathbf{P}} \; \underset{m\times n}{\mathbf{Q}} \; \underset{n\times n}{\mathbf{E}} \tag{3.14}$$

where

- $\mathbf{P}$ *is an* $m \times m$ *permutation matrix,*
- $\mathbf{A}$ *is a nonsingular* $k \times k$ *lower-triangular matrix,*
- $\mathbf{B}$ *ia a* $(m-k) \times k$ *matrix,*
- $\mathbf{E}$ *is an* $n \times n$ *unimodular matrix*

The proof of the theorem is long and tedious. We prefer to give the idea of the proof with a specific example.

Example 3.4. We illustrate Theorem 3.1 in the case $m \times n = 5 \times 4$

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 \\ 11 & 2 & 1 & 14 \\ 14 & 3 & 1 & 18 \\ 3 & 1 & 0 & 4 \end{bmatrix} .$$

First, we apply triangL $[\mathbf{G}]$ to get the lower-triangular form

$$\mathbf{G}_L = \mathbf{G}\,\mathbf{E}_L = \begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 \\ 11 & 2 & 1 & 14 \\ 14 & 3 & 1 & 18 \\ 3 & 1 & 0 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & -5 & -4 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \mathbf{0} \\ 0 & 0 & 0 & \mathbf{0} \\ 11 & 2 & 0 & \mathbf{0} \\ 14 & 3 & 2 & \mathbf{0} \\ 3 & 1 & 2 & \mathbf{0} \end{bmatrix}$$

where the last column is zero, which states that the rank of $\mathbf{G}$ is $k = 3$. Hence, we delete the last column to get the 5×3 matrix

$$\mathbf{C}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 11 & 2 & 0 \\ 14 & 3 & 2 \\ 3 & 1 & 2 \end{bmatrix} .$$

To get the decomposition we search for a 3 minor of rank 3. In this case the number of 3×3 minors is $\binom{5}{3} = 10$. The macro `Minors[m,k]` of `Mathematica` gives the list of all $k \times k$ submatrices of the matrix `m`. Then it is possible to get the vector $\mathbf{c}$ of the determinant of such submatrices. For the matrix $\mathbf{C}_1$ the vector is

$$\mathbf{c} = [0 \quad 0 \quad 0 \quad \mathbf{4} \quad 4 \quad 4 \quad 0 \quad 0 \quad 0 \quad 0]$$

These values are given in *lexicographical order*. In the specific case we have

$$\text{combination}\,[5,3] = \begin{bmatrix} 1 & 1 & 1 & \mathbf{1} & 1 & 1 & 2 & 2 & 2 & 3 \\ 2 & 2 & 2 & \mathbf{3} & 3 & 4 & 3 & 3 & 4 & 4 \\ 3 & 4 & 5 & \mathbf{4} & 5 & 5 & 4 & 5 & 5 & 5 \end{bmatrix}$$

The lexicographical position of the first nonzero determinant in $\mathbf{c}$ allows to establish that the desired minor is formed by the rows **1**, **3**, **4**; in fact

$$\mathbf{A}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 11 & 2 & 0 \\ 14 & 3 & 2 \end{bmatrix} \qquad \text{with } \det(\mathbf{A}_0) = 4 .$$

In such a way we get the matrix $\mathbf{A}_0$ and matrix $\mathbf{B}_0$, formed by the other rows **2**, **5** and the permutation vector $\mathbf{p} = (1,3,4,2,5)$. At this point we get the matrices

$$\mathbf{A}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 11 & 2 & 0 \\ 14 & 3 & 2 \end{bmatrix} \qquad \mathbf{B}_0 = \begin{bmatrix} 0 & 0 & 0 \\ 3 & 1 & 2 \end{bmatrix} \qquad \mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where the permutation matrix $\mathbf{P}$ is identified by the vector $\mathbf{p}$. Hence

$$\mathbf{C}_0 = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{B}_0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 11 & 2 & 0 \\ 14 & 3 & 2 \\ 0 & 0 & 0 \\ 3 & 1 & 2 \end{bmatrix} .$$

In this particular case the permutation does not destroy the triangular form: $\mathbf{A}_0$ is still triangular, but not in the canonical form. Then we apply triangL$[\mathbf{C}_0]$ to get

$$\mathbf{C} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 1 & 2 \\ 0 & 0 & 0 \\ 0 & -1 & 2 \end{bmatrix}$$

with

$$\mathbf{E}_0 = \begin{bmatrix} 1 & 0 & 0 \\ -5 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix}$$

that is,

$$\mathbf{C} = \mathbf{C}_0\,\mathbf{E}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 11 & 2 & 0 \\ 14 & 3 & 2 \\ 0 & 0 & 0 \\ 3 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -5 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 1 & 2 \\ 0 & 0 & 0 \\ 0 & -1 & 2 \end{bmatrix} .$$

After the evaluation of $\mathbf{E}_0^{-1}$ we get the final unimodular matrix

$$\mathbf{E} = \begin{bmatrix} \mathbf{E}_0^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{E}_L$$

$$\mathbf{E} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 5 & 1 & 0 & 0 \\ 4 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & -5 & -4 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 2 \\ 5 & 1 & 0 & 6 \\ 4 & 1 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$

From the previous evaluation we get the desired decomposition

$$\underset{\mathbf{G}}{\begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 \\ 11 & 2 & 1 & 14 \\ 14 & 3 & 1 & 18 \\ 3 & 1 & 0 & 4 \end{bmatrix}} = \underset{\mathbf{P}}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}} \underset{\mathbf{Q}}{\begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & 0 \\ \mathbf{1} & \mathbf{2} & \mathbf{0} & 0 \\ \mathbf{1} & \mathbf{1} & \mathbf{2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 \end{bmatrix}} \underset{\mathbf{E}}{\begin{bmatrix} 1 & 0 & 1 & 2 \\ 5 & 1 & 0 & 6 \\ 4 & 1 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

where the entries in boldface give the matrix **A** inside the matrix **Q**.

Usage

```
{k,comb,rest,p,pP,q,a,b,e} = degdecomposition[g,op]
```

where

- `g` is the given integer matrix,
- `op` is an optional arguments with default value `op = 0`. With `op = 1` the lower triangularization is applied,
- `k` is the rank of the matrix `g`
- `comb` is the vector giving the permutation of the `k` rows of the matrix `a`,
- `rest` is the vector giving the permutation of the `m-k` rows of the matrix `b`,
- `q` the matrix of dimension `m x n` formed by four blocks, where the first block is the matrix `a`
- `a` is the full–rank matrix of dimension `k x k`,
- `b` is the matrix of dimension `m-k x k`.

Hence we get the two possible forms

```
{k,comb,rest,p,pP,q,a,b,e} = degdecomposition[a]
{k,comb,rest,p,pP,q,a,b,e} = degdecomposition[a,1]
```

Note that in the procedure all the limit cases as $m = n$, $k = m$, and also the case $m = n = k = 0$, corresponding to a zero matrix `g`, have been checked for a correct result.

3.4.2 *Points of the projection lattice*

The decomposition of Theorem 3.1 can be used to find the points of the degenerate lattice G, with basis **G**, belonging to a given region $\mathcal{R}$ of $\mathbb{R}^m$. We begin with noting that in (3.14) **E** is unimodular and therefore

$$\mathbf{G}_0 = \mathbf{P} \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{B} & \mathbf{0} \end{bmatrix}$$

is a valid basis of G.

The coordinate permutation

$$\mathbf{u} = \mathbf{P}^{-1}\,\mathbf{t} \tag{3.15}$$

allows to get, from the lattice G with basis $\mathbf{G}_0$, the lattice G_p with basis

$$\mathbf{G}_p = \mathbf{P}^{-1}\,\mathbf{G}_0 = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{B} & \mathbf{0} \end{bmatrix}.$$

Permutation (3.15) is more explicitly

$$\mathbf{u} = (t_{p_1}, \dots, t_{p_m})$$

where $\mathbf{p} = (p_1, \dots, p_m)$ is a permutation of $(1, \dots, m)$, which identifies the permutation matrix $\mathbf{P}$.

Now, if the given region is

$$\mathcal{R} = [t_{1_{min}}, t_{1_{max}}] \times \dots \times [t_{m_{min}}, t_{m_{max}}]$$

the permuted region becomes

$$\mathcal{R}_{\mathbf{p}} = [t_{p_{1_{min}}}, t_{p_{1_{max}}}] \times \dots \times [t_{p_{m_{min}}}, t_{p_{m_{max}}}] \tag{3.16}$$

The problem is now: from the generation of the points of G belonging to the region $\mathcal{R}$, one has to generate the points of G_p belonging to the region $\mathcal{R}_p$; from $\mathbf{u} \in \mathcal{R}_p$ we gets $\mathbf{t} \in \mathcal{R}$ by means of the inverse permutation $\mathbf{t} = \mathbf{P}\,\mathbf{u}$.

The generation of the points of G_p belonging to $\mathcal{R}_p$ can be formulated as follows. A generic point of G_p, $\mathbf{u} = \mathbf{G}_p\,\mathbf{h}$, can be decomposed in the form

$$\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \end{bmatrix}, \qquad \mathbf{h}_1 \in \mathbb{Z}^k$$

where k is the rank of $\mathbf{G}$, that is,

$$\mathbf{u}_1 = \mathbf{A}\,\mathbf{h}_1 \qquad \mathbf{u}_2 = \mathbf{B}\,\mathbf{h}_1 = \mathbf{B}\,\mathbf{A}^{-1}\,\mathbf{u}_1$$

Hence, the generation is achieved from the generation of the points of the lattice A, identified by the basis $\mathbf{A}$, belonging to the region of $\mathbb{R}^k$

$$\mathcal{R}_p(k) = [u_{p_{1_{min}}}, u_{p_{1_{max}}}] \times \dots \times [u_{p_{k_{min}}}, u_{p_{k_{max}}}]$$

Note that, since $\mathbf{A}$ is nonsingular, A turns out to be full-dimensional and can be interpreted as the *projection lattice* of G_p along the hyperplane $\mathbb{R}^k$ of $\mathbb{R}^m$. Once obtained k–dimensional points of A, their coordinates can be completed in the m–dimensional form (of G_p) according to

$$\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2) = (\mathbf{u}_1, \mathbf{B}\mathbf{A}^{-1}\,\mathbf{u}_1), \qquad \mathbf{u}_1 \in A$$

In conclusion, the generation of points a degenerate lattice (kD in mD) is obtained from the generation of the points of a full-dimensional lattice in kD.

3.4.3 The procedure "deglatticepoints"

The procedure is organized as follows:

Given the basis $\mathbf{G}$, $m \times n$ and the mD region $\mathcal{R}$, specified according to (3.16).

1) Find the decomposition of $\mathbf{G}$ given by Theorem 3.1. Hence , one gets the rank k, the nonsingular $k \times k$ $\mathbf{A}$, the matrix $\mathbf{B}$ and the permutation matrix $\mathbf{P}$.

2) The permutation region $\mathcal{R}_p$ and its projection kD $\mathcal{R}_p(k)$ are evaluated.

3) Using the procedure for regular lattices, the pints $\mathbf{u}_1 \in \mathcal{R}_p(k)$ of the lattice $A = \mathbf{A}\mathbb{Z}^k$ are generated.

4) The coordinates are completed according to $\mathbf{u} = (\mathbf{u}_1, \mathbf{B}\mathbf{A}^{-1}\mathbf{u}_1)$, $\mathbf{u}_1 \in \mathcal{R}_p(k)$ and the check if $\mathbf{u} \in \mathcal{R}_p$, *eliminating* the $\mathbf{u} \notin \mathcal{R}_p$.

5) Using the inverse permutation $\mathbf{t} = \mathbf{P}\mathbf{u}$, the points $\mathbf{t} \in \mathcal{R}$ are obtained from the point $\mathbf{u} \in \mathcal{R}_p$. by the inverse permutation $\mathbf{t} = \mathbf{P}\mathbf{u}$.

The above steps are implemented in the fallowing procedure.

Usage

```
deglatticepoints[a_,region_,op_:0] --> p
```

where

- `a` is a lattice basis,
- `region` is the region specified in the form `{rmin1,rmax1,rmin2,rmax2,...,rminm,rmaxm}`
- `op` is an optional argument with default value `op = 0`;
  with `op = 0` it gives all the points in the format `{t1,t2,...tm}`,
  with `op = 1` it gives only the *primitive* in the format `{t1,t2,...tm}`,
  with `op = 2` it gives all the points in the format `{{t1,t2,...tm},z}`,
  where z=0 indicates a nonprimitive point and z=1 indicates a primitive point.
- `p` is the list of the lattice points.

Hence we have the three possible versions

```
p = deglatticepoints[a,region]
p = deglatticepoints[a,region,1]
p = deglatticepoints[a,region,2].
```

Example 3.5. Given the 5×4 matrix seen above

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 \\ 11 & 2 & 1 & 14 \\ 14 & 3 & 1 & 18 \\ 3 & 1 & 0 & 4 \end{bmatrix}$$

which determine a 3D of $\mathbb{R}^5$, and given the region

$$\mathcal{R} = [0,1] \times [0,2] \times [0,3] \times [0,4] \times [0,5] \, .$$

1) the essential data in the decomposition are $m = 5$, $n = 4$, $k = 3$ and the matrices

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 1 & 2 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 2 \end{bmatrix}$$

2) The permuted region and its projection are respectively

$$\mathcal{R}_p = [0,1] \times [0,3] \times [0,4] \times [0,2] \times [0,5]$$
$$\mathcal{R}_p(3) = [0,1] \times [0,3] \times [0,4]$$

3) The 3D points are generated using the basis

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 1 & 2 \end{bmatrix} \, .$$

Then the points of A belonging to $\mathcal{R}_p(3)$ result

$$\mathbf{u}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 2 & 2 & 1 & 1 & 3 & 3 & 3 \\ 0 & 2 & 4 & 1 & 3 & 1 & 3 & 0 & 2 & 4 \end{bmatrix}$$

4) Si passa dalle coordinate (u_1, u_2, u_3) alle coordinate $(u_1, u_2, u_3, u_4, u_5)$

$$(u_4, u_5) = (\alpha u_1 + \beta u_2 + \gamma u_3 \, , \, \delta u_1 + \lambda u_2 + \mu u_3)$$

As seen above, the matrix for calculating the coordinates (u_4, u_5) is

$$\begin{bmatrix} \alpha & \beta & \gamma \\ \delta & \lambda & \mu \end{bmatrix} = \mathbf{B}\mathbf{A}^{-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ -\frac{1}{4} & -\frac{1}{4} & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \end{bmatrix} \, .$$

In such a way one gets the coordinates

$$\begin{bmatrix} u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 4 & -1 & 1 & 0 & 2 & -3 & -1 & 1 \end{bmatrix}$$

and hence the points

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} \quad \Longrightarrow \quad \begin{bmatrix} 0 & 0 & 0 & \mathbf{0} & 0 & 1 & 1 & \mathbf{1} & \mathbf{1} & 1 \\ 0 & 0 & 0 & \mathbf{2} & 2 & 1 & 1 & \mathbf{3} & \mathbf{3} & 3 \\ 0 & 2 & 4 & \mathbf{1} & 3 & 1 & 3 & \mathbf{0} & \mathbf{2} & 4 \\ 0 & 0 & 0 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & 0 \\ 0 & 2 & 4 & -\mathbf{1} & 1 & 0 & 2 & -\mathbf{3} & -\mathbf{1} & 1 \end{bmatrix}$$

Not all the points obtained by *prolating* the coordinates from 3 to 5 belong to the region $\mathcal{R}_p$ (three points have $u_5 \notin [0,5]$).

5) The survived points $\mathbf{u} \in \mathcal{R}_p$ are permuted according to $\mathbf{t} = \mathbf{P}\,\mathbf{u}$ to get

$$\mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \end{bmatrix} \quad \Longrightarrow \quad \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 & 1 & 3 \\ 0 & 2 & 4 & 3 & 1 & 3 & 4 \\ 0 & 2 & 4 & 1 & 0 & 2 & 1 \end{bmatrix}$$

3.5 Unification of the Procedures

The procedures “reglatticepoints” and “deglatticepoints” are unified in the procedure “latticepoints”, where a simple check on the dimensions and on the rank of the basis allows to establish which of the procedure to call: if the basis is square and nonsingular, “reglatticepoints”is used, otherwise “deglatticepoints”.

3.5.1 The procedure “latticepoints”

Usage

```
latticepoints[a_,region_,op_:0] --> p
```

where

- `a` is the lattice basis,
- `region` is the region specified in the form `{rmin1,rmax1,rmin2,rmax2,...,rminm,rmaxm}`
- `op` is an optional argument with default value `op = 0`;
  with `op = 0` it gives all the points in the format `{t1,t2,...tm}`,
  with `op = 1` it gives only the *primitive* in the format `{t1,t2,...tm}`,

with `op = 2` it gives all the points in the format `{{t1,t2,...tm},z}`, where z=0 indicates a nonprimitive point and z=1 indicates a primitive point.

- `p` is the list of the lattice points.

Therefore we have the three possibilities

```
p = latticepoints[a,region]
p = latticepoints[a,region,1]
p = latticepoints[a,region,2].
```

3.6 Smallest Separable Lattice Containing a Lattice

In some applications we have to find the smallest separable lattice G_{min} containing a given lattice G of $\mathbb{Z}(d_1,\ldots,d_m)$.

Theorem 3.2. *The smallest separable lattice G_{min} containing a lattice $G = \mathbf{G}\,\mathbb{Z}^m$ is determined by the* `GCD` *of the rows of the matrix* $\mathbf{G}$.

For instance, with

$$\mathbf{G} = \begin{bmatrix} 6 & 2 & 2 \\ 0 & 4 & 2 \\ 6 & 0 & 9 \end{bmatrix}$$

we find

$$\begin{aligned} \mathrm{GCD}\,[6,2,2] &= 2 \\ \mathrm{GCD}\,[0,4,2] &= 2 \\ \mathrm{GCD}\,[6,0,9] &= 3 \end{aligned} \quad .$$

Hence

$$G_{min} = \mathbb{Z}\,(2,2,3) \quad .$$

La corresponding procedure is

```
leastseplattice[a_]:=
 Table[gcd[ a[[i]] ],{i,Length[a]}]
```

3.7 Demos of Lattice Points

3.7.1 The procedures "writelatticepoints" and "showlatticepoints"

The first procedure writes in TEX the "latticepoints" and the second gives also the graphic representation in the 2D and 3D cases.

Usage

```
writelatticepoints[a,region,op]
showlatticepoints[a,region,op]
```

where

- `a` is a basis of the lattice,
- `region` is the rectangular region specified in the form `{rmin1,rmax1,rmin2,rmax2,...,rminm,rmaxm}`
- `op` is an option with default `op = 0`
  with `op = 0` gives all the points belonging to the region `region`
  with `op = 1` gives the primitive points belonging to the region `region`
  with `op = 2` gives all the points with the primitive points in boldface

Example 3.6. **3D regular lattice**

```
Get["demo2macro.m"]
lab="LP268"
initdemo2[lab]
   d33 = {{2,2,0},
          {0,3,1},
          {1,0,4}}
   region3={0,10,0,10,0,10}
   showlatticepoints[d33,region3]
   completedemo2
```

one gets

Lattice points: run $\boxed{\textbf{LP268}}$ **(May 26, 2011)**

Given an integer matrix (basis), which identify the lattices G and a rectangular region $\mathbf{r}$:

$$\mathbf{G} = \begin{bmatrix} 2 & 2 & 0 \\ 0 & 3 & 1 \\ 1 & 0 & 4 \end{bmatrix} \qquad \mathbf{r} = [0 \quad 8 \quad 0 \quad 8 \quad 0 \quad 8]$$

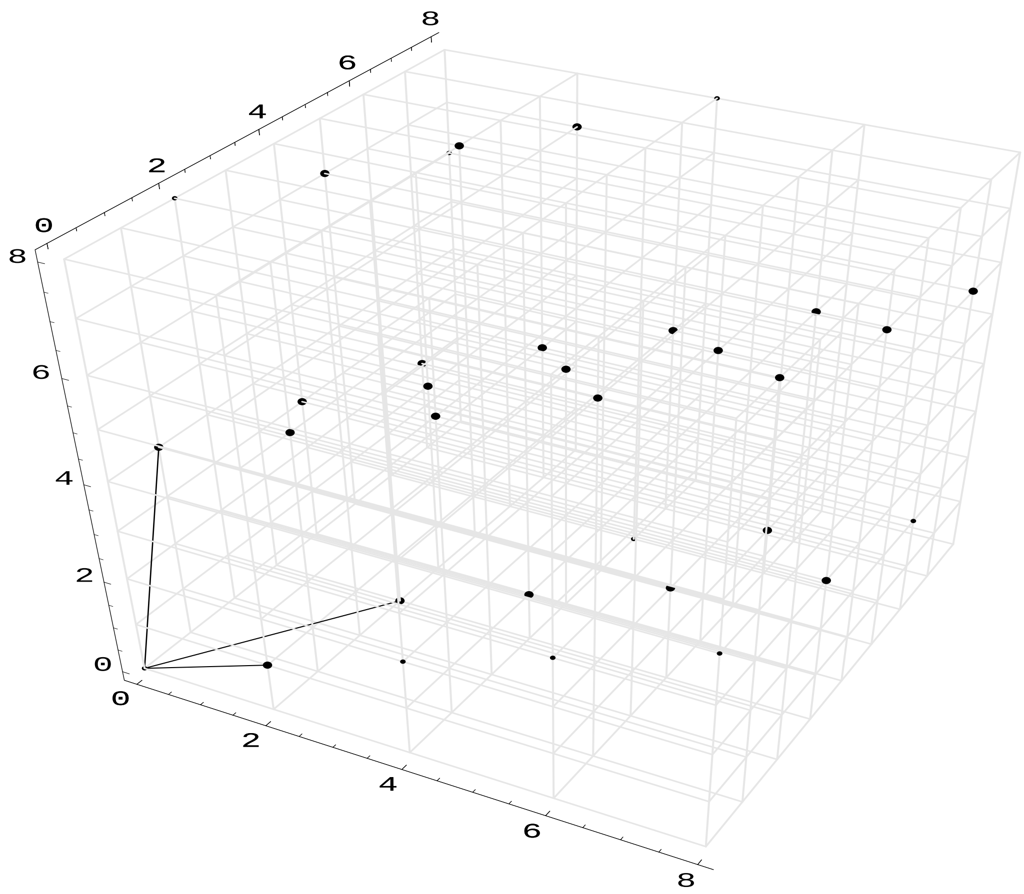
The procedure `writelatticepoints` gives the lattice points generated by $\mathbf{G}$ in the region (extrema included).

$op = 2$: all lattice points with primitive lattice points marked

number of points: 33

$$\begin{bmatrix} 0 & \mathbf{0} & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{2} & 4 & \mathbf{4} & \mathbf{4} & \mathbf{4} & 4 & \mathbf{4} \\ 0 & \mathbf{1} & 2 & \mathbf{4} & \mathbf{5} & \mathbf{7} & 8 & \mathbf{0} & \mathbf{1} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{7} & \mathbf{8} & 0 & \mathbf{1} & \mathbf{3} & \mathbf{4} & 6 & \mathbf{7} \\ 0 & \mathbf{4} & 8 & \mathbf{3} & \mathbf{7} & \mathbf{2} & 6 & \mathbf{1} & \mathbf{5} & \mathbf{0} & \mathbf{4} & \mathbf{8} & \mathbf{3} & \mathbf{7} & 2 & \mathbf{6} & \mathbf{1} & \mathbf{5} & 0 & \mathbf{4} \end{bmatrix}$$

$$\begin{bmatrix} 4 & 6 & \mathbf{6} & \mathbf{6} & \mathbf{6} & \mathbf{6} & \mathbf{6} & 8 & \mathbf{8} & \mathbf{8} & \mathbf{8} & 8 & \mathbf{8} \\ 8 & 0 & \mathbf{1} & \mathbf{3} & \mathbf{4} & \mathbf{6} & \mathbf{7} & 0 & \mathbf{1} & \mathbf{3} & \mathbf{4} & 6 & \mathbf{7} \\ 8 & 3 & \mathbf{7} & \mathbf{2} & \mathbf{6} & \mathbf{1} & \mathbf{5} & 4 & \mathbf{8} & \mathbf{3} & \mathbf{7} & 2 & \mathbf{6} \end{bmatrix}$$



Example 3.7. **lattice with dimension** > 3
Consider the 5D case. With the statements

```
Get["demo2macro.m"]
lab="LP270"
initdemo2[lab]
   a55 = {{2,0,0,1,5},
          {0,3,0,2,7},
          {0,0,2,3,11},
          {0,1,2,3,13},
          {1,3,2,0,3}}
   region5={0,6,0,6,0,6,0,6,0,6}
   writelatticepoints[a55,region5]
   completedemo2
```

one gets

Lattice points: run $\boxed{\textbf{LP270}}$ (May 26, 2011)

Given an integer matrix (basis), which identify the lattices G and a rectangular region $\mathbf{r}$:

$$\mathbf{G} = \begin{bmatrix} 2 & 0 & 0 & 1 & 5 \\ 0 & 3 & 0 & 2 & 7 \\ 0 & 0 & 2 & 3 & 11 \\ 0 & 1 & 2 & 3 & 13 \\ 1 & 3 & 2 & 0 & 3 \end{bmatrix} \qquad \mathbf{r} = [0 \;\; 6 \;\; 0 \;\; 6 \;\; 0 \;\; 6 \;\; 0 \;\; 6 \;\; 0 \;\; 6]$$

The procedure `writelatticepoints` gives the lattice points generated by $\mathbf{G}$ in the region (extrema included).

$op = 2$: all lattice points with primitive lattice points marked

number of points: 152

$$\left[\begin{array}{cccccccccccccccccccc}
0 & \mathbf{0} & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
0 & \mathbf{0} & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & 2 & \mathbf{2} & \mathbf{2} & \mathbf{3} & 3 & \mathbf{3} & \mathbf{4} & \mathbf{4} & \mathbf{4} & \mathbf{5} & \mathbf{5} & \mathbf{5} & \mathbf{5} \\
0 & \mathbf{2} & 4 & 6 & \mathbf{0} & \mathbf{2} & \mathbf{4} & 0 & \mathbf{2} & \mathbf{2} & \mathbf{0} & 0 & \mathbf{2} & \mathbf{0} & \mathbf{2} & \mathbf{6} & \mathbf{0} & \mathbf{0} & \mathbf{4} & \mathbf{6} \\
0 & \mathbf{2} & 4 & 6 & \mathbf{2} & \mathbf{4} & \mathbf{6} & 4 & \mathbf{1} & \mathbf{6} & \mathbf{1} & 6 & \mathbf{3} & \mathbf{3} & \mathbf{5} & \mathbf{1} & \mathbf{0} & \mathbf{5} & \mathbf{1} & \mathbf{3} \\
0 & \mathbf{2} & 4 & 6 & \mathbf{0} & \mathbf{2} & \mathbf{4} & 0 & \mathbf{5} & \mathbf{2} & \mathbf{3} & 0 & \mathbf{5} & \mathbf{3} & \mathbf{5} & \mathbf{2} & \mathbf{6} & \mathbf{3} & \mathbf{0} & \mathbf{2}
\end{array}\right]$$

$$\left[\begin{array}{cccccccccccccccccccc}
0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\
6 & \mathbf{6} & \mathbf{6} & \mathbf{6} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{3} & \mathbf{3} & \mathbf{3} & \mathbf{4} & \mathbf{4} & \mathbf{4} & \mathbf{5} & \mathbf{5} & \mathbf{5} \\
0 & \mathbf{4} & \mathbf{6} & \mathbf{6} & \mathbf{1} & \mathbf{5} & \mathbf{3} & \mathbf{5} & \mathbf{3} & \mathbf{5} & \mathbf{5} & \mathbf{3} & \mathbf{3} & \mathbf{5} & \mathbf{1} & \mathbf{3} & \mathbf{5} & \mathbf{1} & \mathbf{3} & \mathbf{5} \\
2 & \mathbf{3} & \mathbf{0} & \mathbf{5} & \mathbf{5} & \mathbf{1} & \mathbf{1} & \mathbf{3} & \mathbf{3} & \mathbf{0} & \mathbf{5} & \mathbf{0} & \mathbf{5} & \mathbf{2} & \mathbf{0} & \mathbf{2} & \mathbf{4} & \mathbf{2} & \mathbf{4} & \mathbf{6} \\
6 & \mathbf{0} & \mathbf{5} & \mathbf{2} & \mathbf{5} & \mathbf{2} & \mathbf{0} & \mathbf{2} & \mathbf{0} & \mathbf{5} & \mathbf{2} & \mathbf{3} & \mathbf{0} & \mathbf{5} & \mathbf{1} & \mathbf{3} & \mathbf{5} & \mathbf{1} & \mathbf{3} & \mathbf{5}
\end{array}\right]$$

$$\left[\begin{array}{cccccccccccccccccccc}
\mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{2} & 2 & \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{2} \\
\mathbf{6} & \mathbf{6} & \mathbf{6} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{2} & \mathbf{2} & \mathbf{2} & 2 & \mathbf{3} & \mathbf{3} & \mathbf{3} & \mathbf{3} & \mathbf{4} & \mathbf{4} \\
\mathbf{1} & \mathbf{3} & \mathbf{3} & \mathbf{0} & \mathbf{2} & \mathbf{4} & \mathbf{0} & \mathbf{2} & \mathbf{4} & \mathbf{6} & \mathbf{0} & \mathbf{2} & \mathbf{2} & 6 & \mathbf{0} & \mathbf{0} & \mathbf{2} & \mathbf{6} & \mathbf{0} & \mathbf{2} \\
\mathbf{4} & \mathbf{1} & \mathbf{6} & \mathbf{0} & \mathbf{2} & \mathbf{4} & \mathbf{2} & \mathbf{4} & \mathbf{6} & \mathbf{0} & \mathbf{4} & \mathbf{1} & \mathbf{6} & 2 & \mathbf{1} & \mathbf{6} & \mathbf{3} & \mathbf{4} & \mathbf{3} & \mathbf{5} \\
\mathbf{1} & \mathbf{6} & \mathbf{3} & \mathbf{1} & \mathbf{3} & \mathbf{5} & \mathbf{1} & \mathbf{3} & \mathbf{5} & \mathbf{0} & \mathbf{1} & \mathbf{6} & \mathbf{3} & 0 & \mathbf{4} & \mathbf{1} & \mathbf{6} & \mathbf{0} & \mathbf{4} & \mathbf{6}
\end{array}\right]$$

$$\left[\begin{array}{cccccccccccccccccccc}
\mathbf{2} & 2 & \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{2} & 2 & \mathbf{2} & \mathbf{3} & \mathbf{3} & \mathbf{3} & \mathbf{3} & \mathbf{3} & \mathbf{3} & \mathbf{3} & \mathbf{3} & \mathbf{3} & \mathbf{3} & \mathbf{3} & \mathbf{3} \\
\mathbf{4} & 4 & \mathbf{5} & \mathbf{5} & \mathbf{5} & \mathbf{6} & 6 & \mathbf{6} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{3} & \mathbf{3} & \mathbf{3} & \mathbf{4} \\
\mathbf{6} & 6 & \mathbf{0} & \mathbf{4} & \mathbf{6} & \mathbf{4} & 6 & \mathbf{6} & \mathbf{1} & \mathbf{5} & \mathbf{5} & \mathbf{3} & \mathbf{5} & \mathbf{3} & \mathbf{5} & \mathbf{5} & \mathbf{3} & \mathbf{3} & \mathbf{5} & \mathbf{1} \\
\mathbf{1} & 6 & \mathbf{5} & \mathbf{1} & \mathbf{3} & \mathbf{3} & 0 & \mathbf{5} & \mathbf{5} & \mathbf{1} & \mathbf{6} & \mathbf{1} & \mathbf{3} & \mathbf{3} & \mathbf{0} & \mathbf{5} & \mathbf{0} & \mathbf{5} & \mathbf{2} & \mathbf{0} \\
\mathbf{3} & 0 & \mathbf{4} & \mathbf{1} & \mathbf{3} & \mathbf{1} & 6 & \mathbf{3} & \mathbf{6} & \mathbf{3} & \mathbf{0} & \mathbf{1} & \mathbf{3} & \mathbf{1} & \mathbf{6} & \mathbf{3} & \mathbf{4} & \mathbf{1} & \mathbf{6} & \mathbf{2}
\end{array}\right]$$

$$\left[\begin{array}{cccccccccccccccccccc}
\mathbf{3} & \mathbf{3} & \mathbf{3} & \mathbf{3} & \mathbf{3} & \mathbf{3} & \mathbf{3} & 4 & \mathbf{4} & 4 & \mathbf{4} & \mathbf{4} & \mathbf{4} & \mathbf{4} & 4 & \mathbf{4} & \mathbf{4} & \mathbf{4} & \mathbf{4} & \mathbf{4} \\
\mathbf{4} & \mathbf{4} & \mathbf{5} & \mathbf{5} & \mathbf{5} & \mathbf{6} & \mathbf{6} & 0 & \mathbf{0} & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & 2 & \mathbf{2} & \mathbf{2} & \mathbf{3} & \mathbf{3} & \mathbf{3} \\
\mathbf{3} & \mathbf{5} & \mathbf{1} & \mathbf{3} & \mathbf{5} & \mathbf{1} & \mathbf{3} & 0 & \mathbf{2} & 4 & \mathbf{0} & \mathbf{2} & \mathbf{4} & \mathbf{6} & 0 & \mathbf{2} & \mathbf{6} & \mathbf{0} & \mathbf{0} & \mathbf{6} \\
\mathbf{2} & \mathbf{4} & \mathbf{2} & \mathbf{4} & \mathbf{6} & \mathbf{4} & \mathbf{6} & 0 & \mathbf{2} & 4 & \mathbf{2} & \mathbf{4} & \mathbf{6} & \mathbf{0} & 4 & \mathbf{6} & \mathbf{2} & \mathbf{1} & \mathbf{6} & \mathbf{4} \\
\mathbf{4} & \mathbf{6} & \mathbf{2} & \mathbf{4} & \mathbf{6} & \mathbf{2} & \mathbf{4} & 2 & \mathbf{4} & 6 & \mathbf{2} & \mathbf{4} & \mathbf{6} & \mathbf{1} & 2 & \mathbf{4} & \mathbf{1} & \mathbf{5} & \mathbf{2} & \mathbf{1}
\end{array}\right]$$

$$\left[\begin{array}{cccccccccccccccccccc}
\mathbf{4} & \mathbf{4} & \mathbf{4} & \mathbf{4} & \mathbf{4} & \mathbf{4} & \mathbf{4} & \mathbf{4} & \mathbf{4} & \mathbf{5} & \mathbf{5} & \mathbf{5} & \mathbf{5} & \mathbf{5} & \mathbf{5} & \mathbf{5} & \mathbf{5} & \mathbf{5} & \mathbf{5} & \mathbf{5} \\
\mathbf{4} & \mathbf{4} & \mathbf{4} & \mathbf{5} & \mathbf{5} & \mathbf{5} & \mathbf{6} & \mathbf{6} & \mathbf{6} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{3} & \mathbf{3} & \mathbf{3} & \mathbf{4} \\
\mathbf{0} & \mathbf{6} & \mathbf{6} & \mathbf{0} & \mathbf{4} & \mathbf{6} & \mathbf{2} & \mathbf{4} & \mathbf{6} & \mathbf{5} & \mathbf{5} & \mathbf{3} & \mathbf{5} & \mathbf{1} & \mathbf{3} & \mathbf{5} & \mathbf{1} & \mathbf{3} & \mathbf{3} & \mathbf{1} \\
\mathbf{3} & \mathbf{1} & \mathbf{6} & \mathbf{5} & \mathbf{1} & \mathbf{3} & \mathbf{1} & \mathbf{3} & \mathbf{5} & \mathbf{1} & \mathbf{6} & \mathbf{1} & \mathbf{3} & \mathbf{1} & \mathbf{3} & \mathbf{5} & \mathbf{3} & \mathbf{0} & \mathbf{5} & \mathbf{0} \\
\mathbf{5} & \mathbf{4} & \mathbf{1} & \mathbf{5} & \mathbf{2} & \mathbf{4} & \mathbf{0} & \mathbf{2} & \mathbf{4} & \mathbf{4} & \mathbf{1} & \mathbf{2} & \mathbf{4} & \mathbf{0} & \mathbf{2} & \mathbf{4} & \mathbf{0} & \mathbf{5} & \mathbf{2} & \mathbf{3}
\end{array}\right]$$

$$\left[\begin{array}{cccccccccccccccccccc}
\mathbf{5} & \mathbf{5} & \mathbf{5} & \mathbf{5} & \mathbf{5} & \mathbf{5} & \mathbf{5} & 6 & \mathbf{6} & \mathbf{6} & \mathbf{6} & \mathbf{6} & \mathbf{6} & \mathbf{6} & \mathbf{6} & \mathbf{6} & 6 & \mathbf{6} & 6 & \mathbf{6} \\
\mathbf{4} & \mathbf{4} & \mathbf{5} & \mathbf{5} & \mathbf{6} & \mathbf{6} & \mathbf{6} & 0 & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{2} & \mathbf{2} & \mathbf{2} & 2 & \mathbf{3} & 3 & \mathbf{3} \\
\mathbf{1} & \mathbf{3} & \mathbf{1} & \mathbf{3} & \mathbf{1} & \mathbf{3} & \mathbf{5} & 0 & \mathbf{0} & \mathbf{2} & \mathbf{0} & \mathbf{2} & \mathbf{6} & \mathbf{0} & \mathbf{2} & \mathbf{4} & 6 & \mathbf{0} & 0 & \mathbf{4} \\
\mathbf{5} & \mathbf{2} & \mathbf{2} & \mathbf{4} & \mathbf{4} & \mathbf{6} & \mathbf{0} & 0 & \mathbf{5} & \mathbf{2} & \mathbf{2} & \mathbf{4} & \mathbf{0} & \mathbf{4} & \mathbf{6} & \mathbf{0} & 2 & \mathbf{1} & 6 & \mathbf{2} \\
\mathbf{0} & \mathbf{5} & \mathbf{3} & \mathbf{5} & \mathbf{3} & \mathbf{5} & \mathbf{0} & 3 & \mathbf{0} & \mathbf{5} & \mathbf{3} & \mathbf{5} & \mathbf{2} & \mathbf{3} & \mathbf{5} & \mathbf{0} & 2 & \mathbf{6} & 3 & \mathbf{0}
\end{array}\right]$$

$$\left[\begin{array}{cccccccccccc}
\mathbf{6} & \mathbf{6} & \mathbf{6} & \mathbf{6} & 6 & \mathbf{6} & \mathbf{6} & \mathbf{6} & \mathbf{6} & \mathbf{6} & \mathbf{6} & \mathbf{6} \\
\mathbf{3} & \mathbf{4} & \mathbf{4} & \mathbf{4} & 4 & \mathbf{5} & \mathbf{5} & \mathbf{5} & \mathbf{5} & \mathbf{6} & \mathbf{6} & \mathbf{6} \\
\mathbf{6} & \mathbf{0} & \mathbf{4} & \mathbf{6} & 6 & \mathbf{0} & \mathbf{4} & \mathbf{4} & \mathbf{6} & \mathbf{2} & \mathbf{4} & \mathbf{6} \\
\mathbf{4} & \mathbf{3} & \mathbf{4} & \mathbf{1} & 6 & \mathbf{5} & \mathbf{1} & \mathbf{6} & \mathbf{3} & \mathbf{1} & \mathbf{3} & \mathbf{5} \\
\mathbf{2} & \mathbf{6} & \mathbf{0} & \mathbf{5} & 2 & \mathbf{6} & \mathbf{3} & \mathbf{0} & \mathbf{5} & \mathbf{1} & \mathbf{3} & \mathbf{5}
\end{array}\right]$$

Example 3.8. **2D degenerate lattice**
Consider a 1D lattice in 2D

```
Get["demo2macro.m"]
lab="LP272"
initdemo2[lab]
   s22 = {{3, 0},
          {1, 0}}
   region2={0,30,0,15}
   showlatticepoints[s22,region2,2]
   completedemo2
```

one gets

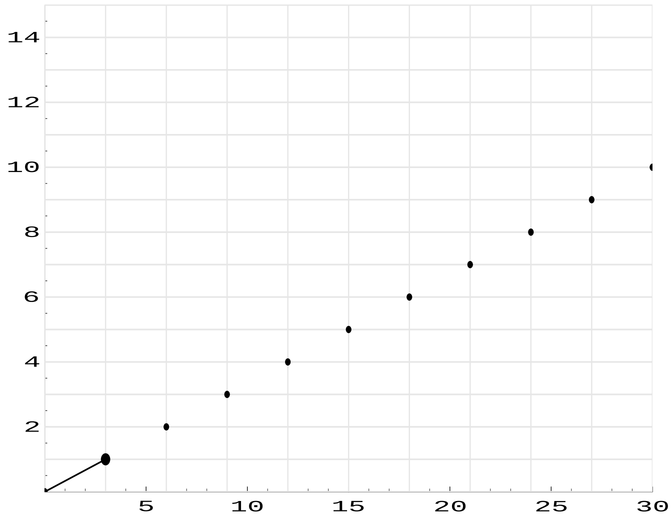
Lattice points: **run** $\boxed{\textbf{LP272}}$ **(May 26, 2011)**
Given an integer matrix (basis), which identify the lattices G and a rectangular region $\mathbf{r}$:

$$\mathbf{G} = \begin{bmatrix} 3 & 0 \\ 1 & 0 \end{bmatrix} \qquad \mathbf{r} = \begin{bmatrix} 0 & 30 & 0 & 15 \end{bmatrix}$$

The procedure `writelatticepoints` gives the lattice points generated by **G** in the region (extrema included).

$op = 2$: all lattice points with primitive lattice points marked
number of points: 11

$$\begin{bmatrix} 0 & \mathbf{3} & 6 & 9 & 12 & 15 & 18 & 21 & 24 & 27 & 30 \\ 0 & \mathbf{1} & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{bmatrix}$$



Example 3.9. **3D degenerate lattice**
Consider a 1D lattice in 2D

```
Get["demo2macro.m"]
lab="LP274"
initdemo2[lab]
   ae33 = {{6,4,4},
           {3,2,2},
           {0,0,9}}
   region32={0,20,0,20,0,20}

   showlatticepoints[ae33,region32,2]
```

The rank of ae33 is 2 and therefore we have a 2D lattice in 3D

Lattice points: run $\boxed{\textbf{LP274}}$ **(May 26, 2011)**
Given an integer matrix (basis), which identify the lattices G and a rectangular region $\mathbf{r}$:

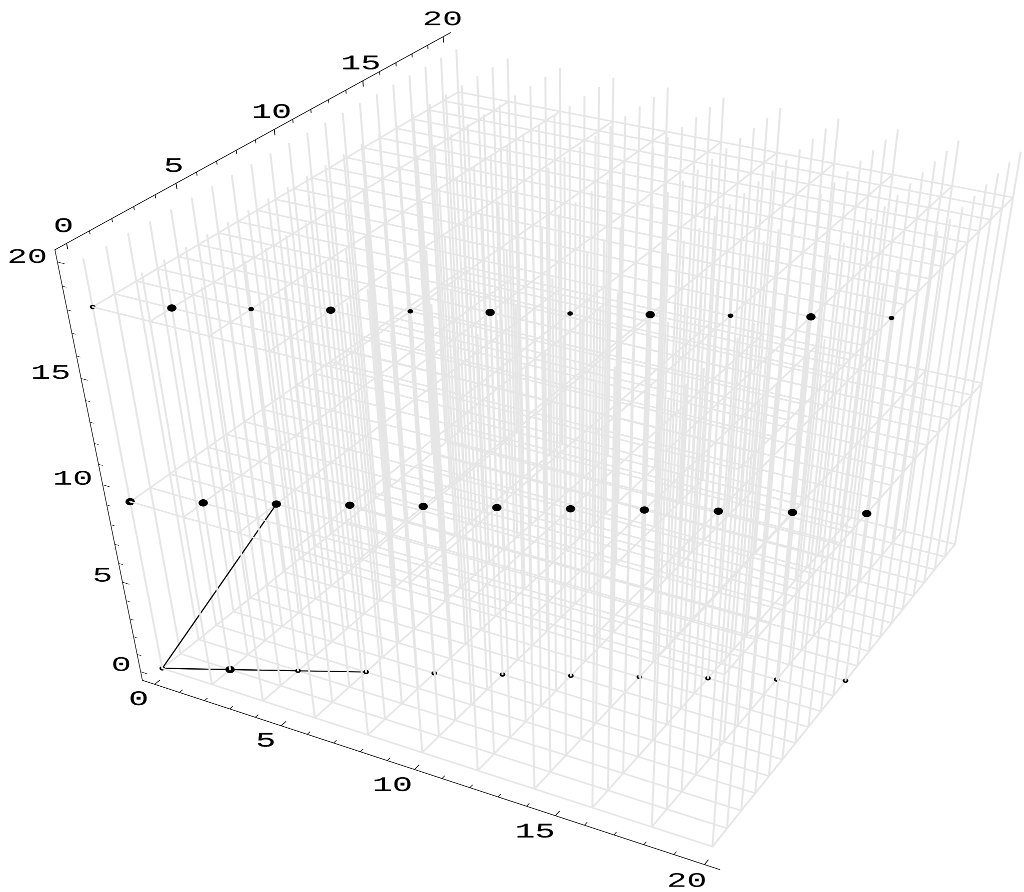
$$\mathbf{G} = \begin{bmatrix} 6 & 4 & 4 \\ 3 & 2 & 2 \\ 0 & 0 & 9 \end{bmatrix} \qquad \mathbf{r} = [0 \quad 20 \quad 0 \quad 20 \quad 0 \quad 20]$$

The procedure `writelatticepoints` gives the lattice points generated by $\mathbf{G}$ in the region (extrema included).

$op = 2$: all lattice points with primitive lattice points marked
number of points: 33

$$\begin{bmatrix} 0 & \mathbf{0} & 0 & \mathbf{2} & \mathbf{2} & \mathbf{2} & 4 & \mathbf{4} & 4 & 6 & \mathbf{6} & \mathbf{6} & 8 & \mathbf{8} & 8 & 10 & \mathbf{10} & \mathbf{10} & 12 & \mathbf{12} \\ 0 & \mathbf{0} & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & 2 & \mathbf{2} & 2 & 3 & \mathbf{3} & \mathbf{3} & 4 & \mathbf{4} & 4 & 5 & \mathbf{5} & \mathbf{5} & 6 & \mathbf{6} \\ 0 & \mathbf{9} & 18 & \mathbf{0} & \mathbf{9} & \mathbf{18} & 0 & \mathbf{9} & 18 & 0 & \mathbf{9} & \mathbf{18} & 0 & \mathbf{9} & 18 & 0 & \mathbf{9} & \mathbf{18} & 0 & \mathbf{9} \end{bmatrix}$$

$$\begin{bmatrix} 12 & 14 & \mathbf{14} & \mathbf{14} & 16 & \mathbf{16} & 16 & 18 & \mathbf{18} & \mathbf{18} & 20 & \mathbf{20} & 20 \\ 6 & 7 & \mathbf{7} & \mathbf{7} & 8 & \mathbf{8} & 8 & 9 & \mathbf{9} & \mathbf{9} & 10 & \mathbf{10} & 10 \\ 18 & 0 & \mathbf{9} & \mathbf{18} & 0 & \mathbf{9} & 18 & 0 & \mathbf{9} & \mathbf{18} & 0 & \mathbf{9} & 18 \end{bmatrix}$$



Example 3.10. **5D degenerate lattice**

```
Get["demo2macro.m"]
lab="LP276"
initdemo2[lab]
   e54 = {{2,3,0,1},
          {0,6,0,2},
          {1,2,2,0},
          {3,5,2,1},
          {1,2,2,0}}
   region6={0,6,0,6,0,6,0,6,0,6}

   writelatticepoints[e54,region6,2]
(************************************)
```

The rank of e54 is 3 and therefore we have a 3D lattice in 5D.

Lattice points: run $\boxed{\textbf{LP276}}$ (May 26, 2011)

Given an integer matrix (basis), which identify the lattices G and a rectangular region $\mathbf{r}$:

$$\mathbf{G} = \begin{bmatrix} 2 & 3 & 0 & 1 \\ 0 & 6 & 0 & 2 \\ 1 & 2 & 2 & 0 \\ 3 & 5 & 2 & 1 \\ 1 & 2 & 2 & 0 \end{bmatrix} \qquad \mathbf{r} = \begin{bmatrix} 0 & 6 & 0 & 6 & 0 & 6 & 0 & 6 & 0 & 6 \end{bmatrix}$$

The procedure `writelatticepoints` gives the lattice points generated by $\mathbf{G}$ in the region (extrema included).

$op = 2$: all lattice points with primitive lattice points marked

number of points: 28

$$\begin{bmatrix} 0 & \mathbf{0} & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{2} & \mathbf{2} & 2 & \mathbf{2} & 2 & \mathbf{3} & \mathbf{3} \\ 0 & \mathbf{0} & 0 & 0 & \mathbf{4} & \mathbf{4} & \mathbf{4} & \mathbf{2} & \mathbf{2} & \mathbf{2} & \mathbf{6} & \mathbf{6} & \mathbf{6} & \mathbf{0} & \mathbf{0} & 4 & \mathbf{4} & 4 & \mathbf{2} & \mathbf{2} \\ 0 & \mathbf{2} & 4 & 6 & \mathbf{1} & \mathbf{3} & \mathbf{5} & \mathbf{0} & \mathbf{2} & \mathbf{4} & \mathbf{1} & \mathbf{3} & \mathbf{5} & \mathbf{1} & \mathbf{3} & 0 & \mathbf{2} & 4 & \mathbf{1} & \mathbf{3} \\ 0 & \mathbf{2} & 4 & 6 & \mathbf{1} & \mathbf{3} & \mathbf{5} & \mathbf{1} & \mathbf{3} & \mathbf{5} & \mathbf{2} & \mathbf{4} & \mathbf{6} & \mathbf{3} & \mathbf{5} & 2 & \mathbf{4} & 6 & \mathbf{4} & \mathbf{6} \\ 0 & \mathbf{2} & 4 & 6 & \mathbf{1} & \mathbf{3} & \mathbf{5} & \mathbf{0} & \mathbf{2} & \mathbf{4} & \mathbf{1} & \mathbf{3} & \mathbf{5} & \mathbf{1} & \mathbf{3} & 0 & \mathbf{2} & 4 & \mathbf{1} & \mathbf{3} \end{bmatrix}$$

$$\begin{bmatrix} 3 & \mathbf{3} & \mathbf{4} & 4 & \mathbf{4} & \mathbf{5} & \mathbf{5} & \mathbf{6} \\ 6 & \mathbf{6} & \mathbf{0} & 0 & \mathbf{4} & \mathbf{2} & \mathbf{6} & \mathbf{4} \\ 0 & \mathbf{2} & \mathbf{0} & 2 & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ 3 & \mathbf{5} & \mathbf{4} & 6 & \mathbf{5} & \mathbf{5} & \mathbf{6} & \mathbf{6} \\ 0 & \mathbf{2} & \mathbf{0} & 2 & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{bmatrix}$$

Chapter 4
Sum and Intersection of Lattices

This chapter deals with the `Mathematica` implementation of the sum $J+K$ and of the intersection $J \cap K$of two mD lattices J and K. Given the bases $\mathbf{J}$ and $\mathbf{K}$ of the lattices, we find the bases of their sum and intersection. The degenerate case is also considered.

Procedures of this chapter

- `lcrmGCLD`: gives the sum and the intersection of two lattices specified by integer matrices (including the case of degenerate lattices);
- `writelcrmGCLD`: write in TEX the sum and the intersection of two lattices specified by integer matrices;
- `showlcrmGCLD`: illustrates the sum and the intersection of two lattices specified by integer matrices in the cases 2D and 3D

4.1 Summary of the Theory

From the *Unified Signal Theory* in particular from Chapter 3 and Chapter 16, we recall the essential things about the sum and the intersection.

1) Given two Abelian groups J and K, and in particular two lattices, the sum is given by

$$J+K = \{j+k \mid j \in J, k \in K\},$$

while the intersection is given by the usual set operation

$$J \cap K = \{u \mid u \in J, u \in K\} .$$

2) If J and K are LCA groups of $\mathbb{R}^m$, their sum $J+K$ is not always an LCA group, while the intersection $J \cap K$ is always an LCA group but may have a dimension $< m$.

3) In particular, in the class $\mathcal{L}(\mathbb{Z}^m)$, if J and K are full-dimensional, also $J+K$ is full-dimensional, that is,

$$J,K \in \mathcal{L}_m(\mathbb{Z}^m) \Longrightarrow J+K,\, J\cap K \in \mathcal{L}_m(\mathbb{Z}^m) \tag{4.1}$$

4) In general, for the dimensions the following relation holds

$$\dim(J+K) + \dim(J\cap K) = \dim J + \dim K$$

where as

$$\dim(J\cap K) \le \dim J,\, \dim K \le \dim(J+K) \le m$$

Relation (4.1) allows to check the dimensions in the evaluation of the sum and intersection.

In Section 16.8 of the *Unified Signal Theory* we have seen that the evaluation of the sum and intersection can be limited to the sublattices of $\mathbb{Z}^m$, whose bases are given by the class $\mathcal{I}_n$ of nonsingular integer matrices. Also, to handle the sum and intersection in $\mathcal{L}_m(\mathbb{Z}^m)$ we have to extend the concepts of GCD and lcm to integer matrices with a long journey on the algebra of integer matrices. The final statement is:

Theorem 4.1. *Let M and N be two lattices of $\mathcal{L}_m(\mathbb{Z}^m)$ with bases* $\mathbf{M}$ *and* $\mathbf{N}$*, respectively. To find the sum $M+N$ and the intersection $M\cap N$*

1) *Compose the $m\times 2m$ matrix $[\mathbf{M}|\mathbf{N}]$.*
2) *Evaluate the Hermite lower-triangular form of $[\mathbf{M}|\mathbf{N}]$. This form has the structure $[\mathbf{L}_0|\mathbf{0}]$ and is obtained by a right multiplication of $[\mathbf{M}|\mathbf{N}]$ by a $2m\times 2m$ unimodular matrix* $\mathbf{E}$

$$[\mathbf{M}|\mathbf{N}]\,\mathbf{E} = [\mathbf{L}_0|\mathbf{0}]\,, \qquad \mathbf{E} = \begin{bmatrix} \mathbf{E}_{11} & \mathbf{E}_{12} \\ \mathbf{E}_{21} & \mathbf{E}_{22} \end{bmatrix} \tag{4.2}$$

where $\mathbf{E}_{ij}$ are $m\times m$ matrices.

Then, $\mathbf{L}_0$ is a basis of $M+N$ and $\mathbf{R}_0 = \mathbf{N}\mathbf{E}_{22}$ is a basis of $M\cap N$.

At the end of the calculation we can check the determinant identity

$$\mathrm{d}(M+N)\,\mathrm{d}(M\cap N) = \mathrm{d}(M)\,\mathrm{d}(N)\,. \tag{4.3}$$

We finally recall the relation linking the intersection to the reciprocal of the sum

$$J\cap K = (J^\star + K^\star)^\star\,.$$

4.1.1 *The procedure "lcrmGCLD"*

Usage

```
{r,s} =lcrmGCLD[a_,b_,adj_:0]
```

where

- `a` and `b` are the bases of the lattices,
- `r` and `s` are the bases of the sum and of the intersection,

Example 4.1. (**4D regular lattices**) With the `Mathematica` statements

```
Get["demo3macro.m"]
lab="SI218"
initdemo3[lab]
a44={{1,0,3,-2},{2,4,-1,5},{-1,5,0,3},{1,1,3,0}}
b44={{0,2,3,-1},{2,0,6,0},{-1,1,0,3},{1,0,3,7}}
writelcrmGCLD[a44,b44]
completedemo3
```

TEX and math give:

Sum and Intersection of lattices: run $\boxed{\textbf{SI218}}$ (May 26, 2011)
Given two integer matrices (bases) **J** and **K**, which identify the lattices J and K:

$$\mathbf{J} = \begin{bmatrix} 1 & 0 & 3 & -2 \\ 2 & 4 & -1 & 5 \\ -1 & 5 & 0 & 3 \\ 1 & 1 & 3 & 0 \end{bmatrix} \qquad \mathbf{K} = \begin{bmatrix} 0 & 2 & 3 & -1 \\ 2 & 0 & 6 & 0 \\ -1 & 1 & 0 & 3 \\ 1 & 0 & 3 & 7 \end{bmatrix}$$

The procedure `writelcrmGCLD` find a basis **C** of the sum $J+K$ and a basis **D** of the intersection $J \cap K$

$$\mathbf{C} = \mathsf{GCD}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ -1 & 13 & -1 & 0 \\ 1 & 2 & 19 & -1 \end{bmatrix}$$

$$\mathbf{D} = \mathsf{lcm}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 0 & 24 & 573 & -4521 \\ 10 & 4204 & 99870 & -791904 \\ -5 & -2090 & -49647 & 393695 \\ 5 & 2102 & 49935 & -395945 \end{bmatrix}$$

Upper-triangular forms

$$\mathbf{J_u} = \begin{bmatrix} 5 & 1 & 2 & 3 \\ 0 & 4 & 1 & 3 \\ 0 & 0 & 3 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{K_u} = \begin{bmatrix} 0 & 2 & 3 & -1 \\ 2 & 0 & 6 & 0 \\ -1 & 1 & 0 & 3 \\ 1 & 0 & 3 & 7 \end{bmatrix}$$

$$\mathbf{C_u} = \mathsf{GCD}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{D_u} = \mathsf{lcm}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 15 & 12 & 12 & 3 \\ 0 & 28 & 14 & 16 \\ 0 & 0 & 6 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Determinant check

$$d(\mathbf{J})\,d(\mathbf{K}) = d(\mathbf{C})\,d(\mathbf{D}) \qquad 60 \cdot 42 = 1 \cdot 2520 \quad (2520 = 2520)$$

Example 4.2. (**3D degenerate lattices**) With the demo

```
Get["demo3macro.m"]
lab="SI211"
initdemo3[lab]
m33={{5,2,0},
    {1,0,0},     (* reduced rank *)
    {0,-1,0}}
n33={{0,2,6},
    {0,0,2},
    {0,-1,1}}
writelcrmGCLD[m33,n33]
completedemo3
```

one gets

Sum and Intersection of lattices: run $\boxed{\textbf{SI211}}$ **(May 26, 2011)**

Given two integer matrices (bases) **J** and **K**, which identify the lattices J and K:

$$\mathbf{J} = \begin{bmatrix} 5 & 2 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \qquad \mathbf{K} = \begin{bmatrix} 0 & 2 & 6 \\ 0 & 0 & 2 \\ 0 & -1 & 1 \end{bmatrix}$$

The procedure `writelcrmGCLD` find a basis **C** of the sum $J+K$ and a basis **D** of the intersection $J \cap K$

$$\mathbf{C} = \mathsf{GCD}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -2 & 0 \\ 2 & -5 & -1 \end{bmatrix}$$

$$\mathbf{D} = \mathsf{lcm}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 0 & 2 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

Upper-triangular forms

$$\mathbf{J_u} = \begin{bmatrix} 0 & 5 & -2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{K_u} = \begin{bmatrix} 0 & 2 & 6 \\ 0 & 0 & 2 \\ 0 & -1 & 1 \end{bmatrix}$$

$$\mathbf{C_u} = \mathsf{GCD}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{D_u} = \mathsf{lcm}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 0 & 0 & -2 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Determinant check

$$d(\mathbf{J})\,d(\mathbf{K}) = d(\mathbf{C})\,d(\mathbf{D}) \qquad 0\cdot 0 = 2\cdot 0 \quad (0=0)$$

4.2 Demos with Illustration

Example 4.3. (**2D full lattices**)

```
Get["demo3macro.m"]
lab="SI112"
initdemo3[lab]
mm=20
nn1=20
quanto=25
marca=8
region2D={mm,nn1,quanto,marca}
a22={{2,1},
     {0,1}}
b22={{6,2},
     {0,1}}
 showlcrmGCLD[a22,b22,region2D,1]
completedemo3
```

Sum and Intersection of lattices: run $\boxed{\text{SI112}}$ (May 26, 2011)
Given two integer matrices (bases) **J** and **K**, which identify the lattices J and K:

$$\mathbf{J} = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} \qquad \mathbf{K} = \begin{bmatrix} 6 & 2 \\ 0 & 1 \end{bmatrix}$$

The procedure `writelcrmGCLD` find a basis **C** of the sum $J+K$ and a basis **D** of the intersection $J \cap K$

$$\mathbf{C} = \mathsf{GCD}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{D} = \mathsf{lcm}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 6 & 4 \\ 0 & 2 \end{bmatrix}$$

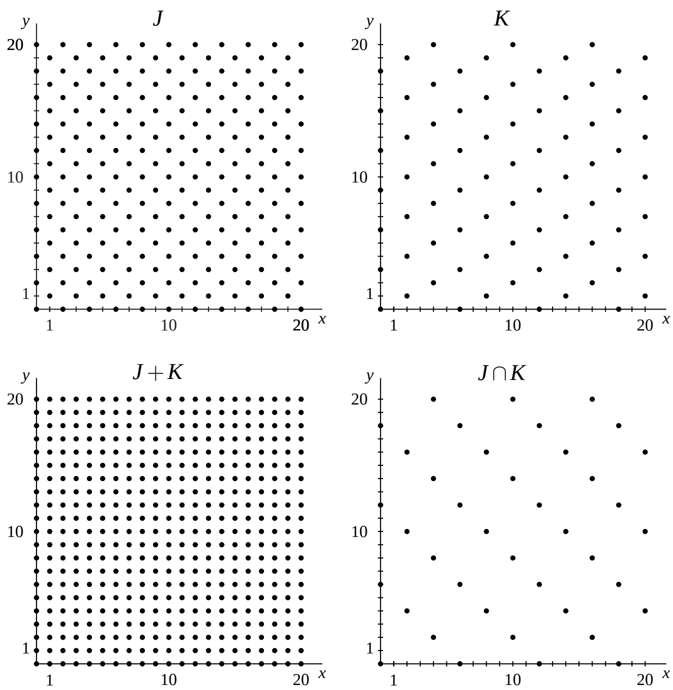
Upper-triangular forms

$$\mathbf{J_u} = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} \qquad \mathbf{K_u} = \begin{bmatrix} 6 & 2 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{C_u} = \mathsf{GCD}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \mathbf{D_u} = \mathsf{lcm}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 6 & 4 \\ 0 & 2 \end{bmatrix}$$

Determinant check

$$d(\mathbf{J})\,d(\mathbf{K}) = d(\mathbf{C})\,d(\mathbf{D}) \qquad 2 \cdot 6 = 1 \cdot 12 \quad (12 = 12)$$



Example 4.4. (**2D degenerate lattices**)

```
Get["demo3macro.m"]
lab="SI113"
initdemo3[lab]
mm=20
nn1=20
quanto=25
marca=8
region2D={mm,nn1,quanto,marca}
ee3={{1,0},
     {1,0}}
ee4={{0,1},
     {0,2}}
showlcrmGCLD[ee3,ee4,region2D,2]
completedemo3
```

Sum and Intersection of lattices: **run** $\boxed{\textbf{SI113}}$ **(May 26, 2011)**

Given two integer matrices (bases) **J** and **K**, which identify the lattices J and K:

$$\mathbf{J} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \qquad \mathbf{K} = \begin{bmatrix} 0 & 1 \\ 0 & 2 \end{bmatrix}$$

The procedure `writelcrmGCLD` find a basis **C** of the sum $J+K$ and a basis **D** of the intersection $J \cap K$

$$\mathbf{C} = \mathsf{GCD}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{D} = \mathsf{lcm}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

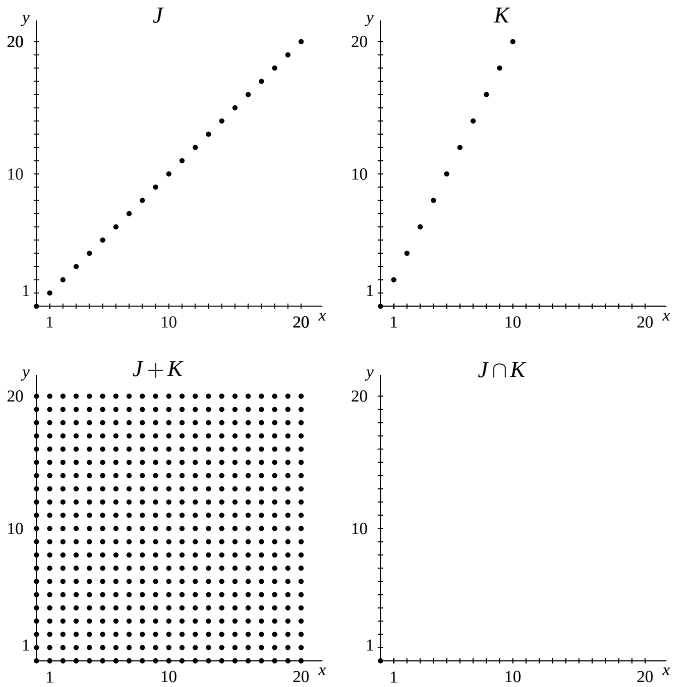
Upper-triangular forms

$$\mathbf{J_u} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \qquad \mathbf{K_u} = \begin{bmatrix} 0 & 1 \\ 0 & 2 \end{bmatrix}$$

$$\mathbf{C_u} = \mathsf{GCD}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \mathbf{D_u} = \mathsf{lcm}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Determinant check

$$d(\mathbf{J})\,d(\mathbf{K}) = d(\mathbf{C})\,d(\mathbf{D}) \qquad 0 \cdot 0 = 1 \cdot 0 \quad (0 = 0)$$



Example 4.5. (**3D full lattices**)

```
Get["demo3macro.m"]
lab="SI120"
initdemo3[lab]
region3D={0,15,0,15,0,15}
c33={{5,2,0},
    {1,0,1},
    {0,-1,8}}
d33={{5,1,3},
    {1,4,2},
    {0,3,1}}

showlcrmGCLD[c33,d33,region3D,1]
completedemo3
```

Sum and Intersection of lattices: run $\boxed{\text{SI120}}$ (May 26, 2011)

Given two integer matrices (bases) **J** and **K**, which identify the lattices J and K:

$$\mathbf{J} = \begin{bmatrix} 5 & 2 & 0 \\ 1 & 0 & 1 \\ 0 & -1 & 8 \end{bmatrix} \qquad \mathbf{K} = \begin{bmatrix} 5 & 1 & 3 \\ 1 & 4 & 2 \\ 0 & 3 & 1 \end{bmatrix}$$

The procedure `writelcrmGCLD` find a basis **C** of the sum $J+K$ and a basis **D** of the intersection $J \cap K$

$$\mathbf{C} = \mathsf{GCD}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{D} = \mathsf{lcm}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 22 & 5 & 9 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Upper-triangular forms

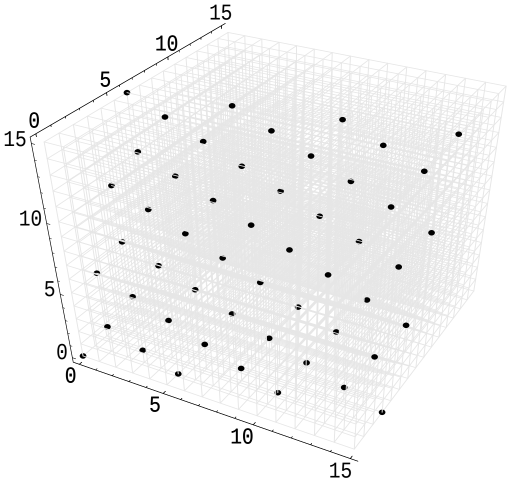
$$\mathbf{J_u} = \begin{bmatrix} 11 & 5 & 9 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{K_u} = \begin{bmatrix} 5 & 1 & 3 \\ 1 & 4 & 2 \\ 0 & 3 & 1 \end{bmatrix}$$

$$\mathbf{C_u} = \mathsf{GCD}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{D_u} = \mathsf{lcm}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 22 & 5 & 9 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

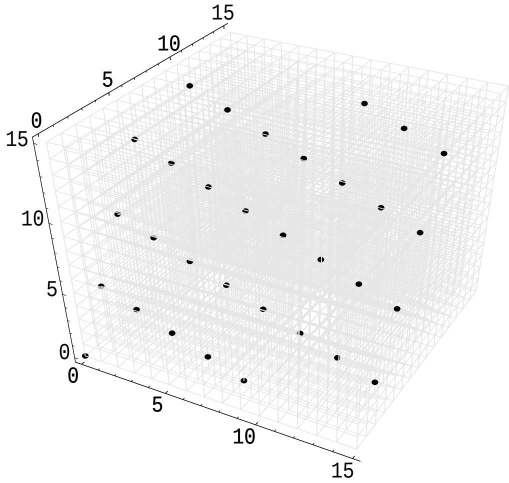
Determinant check

$$d(\mathbf{J})\,d(\mathbf{K}) = d(\mathbf{C})\,d(\mathbf{D}) \qquad 11 \cdot 2 = 1 \cdot 22 \quad (22 = 22)$$

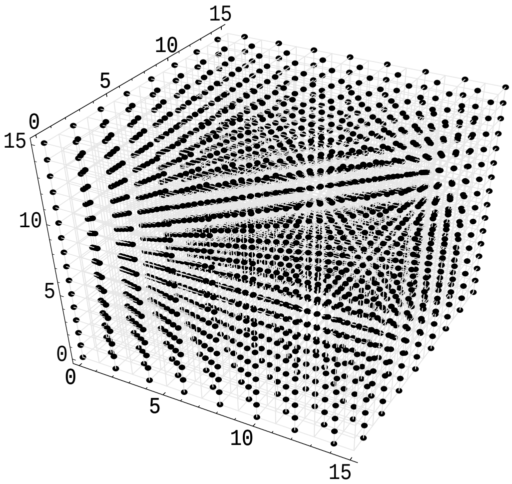
first lattice J



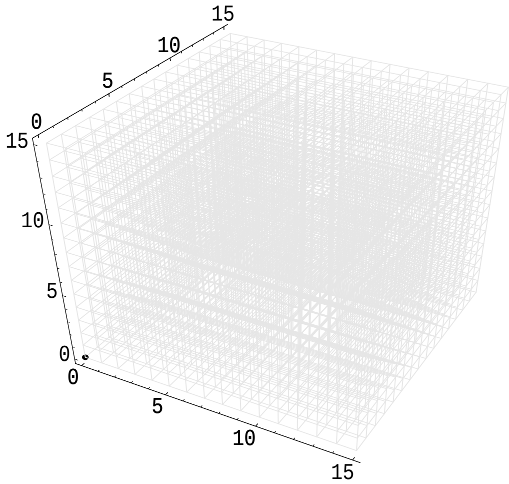
second lattice K



sum $J+K$



intersection $J \cap K$



Example 4.6. (**reduced-dimensional lattices in 3D**)

```
Get["demo3macro.m"]
lab="SI122"
initdemo3[lab]
region3D={0,15,0,15,0,15}
m33={{5,2,0},    (* coppia m,n interessante *)
    {1,0,0},     (* rango ridotto *)
    {0,-1,0}}
n33={{0,2,6},
    {0,0,2},
    {0,-1,1}}
showlcrmGCLD[m33,n33,region3D,1]
completedemo3
```

Sum and Intersection of lattices: run $\boxed{\textbf{SI122}}$ **(May 26, 2011)**

Given two integer matrices (bases) **J** and **K**, which identify the lattices J and K:

$$\mathbf{J} = \begin{bmatrix} 5 & 2 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \qquad \mathbf{K} = \begin{bmatrix} 0 & 2 & 6 \\ 0 & 0 & 2 \\ 0 & -1 & 1 \end{bmatrix}$$

The procedure `writelcrmGCLD` find a basis **C** of the sum $J+K$ and a basis **D** of the intersection $J \cap K$

$$\mathbf{C} = \mathsf{GCD}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{D} = \mathsf{lcm}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 0 & 0 & -2 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Upper-triangular forms

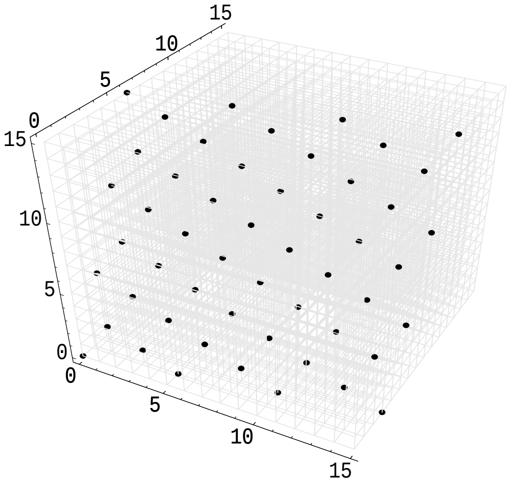
$$\mathbf{J_u} = \begin{bmatrix} 0 & 5 & -2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{K_u} = \begin{bmatrix} 0 & 2 & 6 \\ 0 & 0 & 2 \\ 0 & -1 & 1 \end{bmatrix}$$

$$\mathbf{C_u} = \mathsf{GCD}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{D_u} = \mathsf{lcm}(\mathbf{J},\mathbf{K}) = \begin{bmatrix} 0 & 0 & -2 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

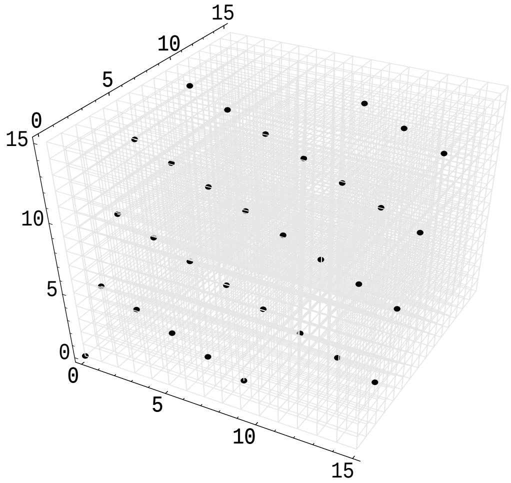
Determinant check

$$d(\mathbf{J})\,d(\mathbf{K}) = d(\mathbf{C})\,d(\mathbf{D}) \qquad 0 \cdot 0 = 2 \cdot 0 \quad (0 = 0)$$

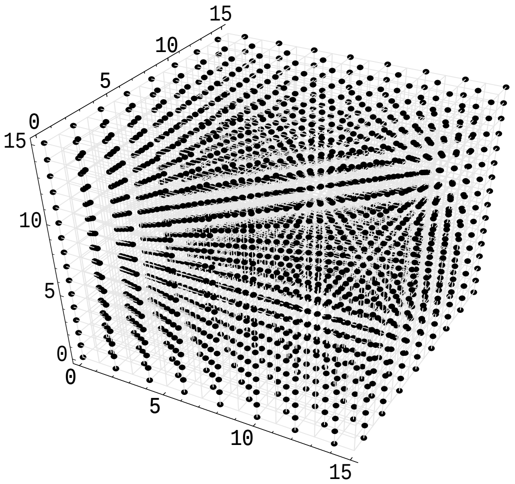
first lattice J



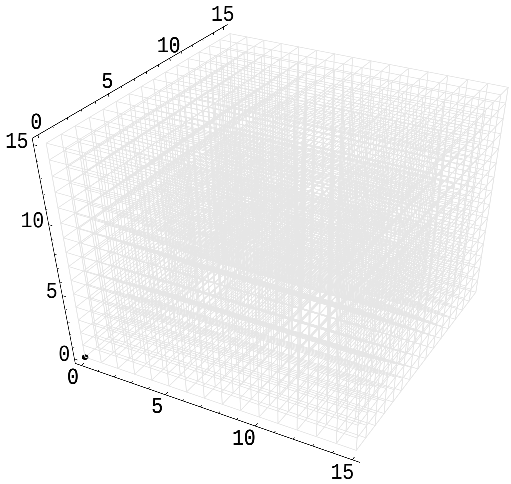
second lattice K



sum $J + K$



intersection $J \cap K$



Chapter 5
Basis Alignment

The chapter deals with the `Mathematica` implementation of the alignment of the bases of a lattice and a sublattice. The theory of this topic is developed in Chapter 16 of the *Unified Signal Theory*.

Procedures of the chapter

- `basealignment`: gives the aligned bases of a lattice and its sublattice(Sect. 5.2)
- `writealignment`: write (in TEX) the matrices that specify the lattice-sublattice, their lower triangular forms, their Smith decomposition, the aligned bases and the diagonal matrix linking the two aligned bases (Sect. 5.3;
- `showalignment`: write (in TEX) the same matrices of "writealignment" and also gives the graphical representation in the cases 2D and 3D (Sect. 5.3).

5.1 Theory

From Chapter 16 of the *Unified Signal Theory* we recall:

5.1.1 Basis alignment of a lattice and a sublattice

Let G be a lattice and J a sublattice. The corresponding bases, $\mathbf{G}_0 = [\mathbf{g}_1 \cdots \mathbf{g}_m]$ and $\mathbf{J}_0 = [\mathbf{j}_1, \dots, \mathbf{J}_m]$, are *aligned* if they verify the condition

$$\mathbf{J}_0 = \mathbf{G}_0 \boldsymbol{\Delta} , \qquad \boldsymbol{\Delta} = \mathrm{diag}(\delta_1, \dots, \delta_m) , \tag{5.1}$$

where δ_k are naturals. For the basis vectors the alignment condition becomes

$$\mathbf{j}_1 = \delta_1 \mathbf{g}_1, \dots, \mathbf{j}_m = \delta_m \mathbf{g}_m . \tag{5.1a}$$

The basis alignment is always possible. In fact:

Theorem 5.1. *Let $G = \mathbf{G}\mathbb{Z}^m$ and $J = \mathbf{J}\mathbb{Z}^m$, with J sublattice of J. Let $\mathbf{J} = \mathbf{G}\mathbf{H}$ with $\mathbf{H} \in \mathfrak{I}_m$. Then, the Smith decomposition $\mathbf{H} = \mathbf{E}_1 \Delta \mathbf{E}_2$ allows to define the alignment bases as*

$$\mathbf{G}_0 = \mathbf{G}\,\mathbf{E}_1\,, \qquad \mathbf{J}_0 = \mathbf{J}\,\mathbf{E}_2^{-1}\,. \tag{5.2}$$

The basis alignment gives a detailed information about the *density reduction* in passing from a lattice to its sublattice. From the generic basis relation $\mathbf{J} = \mathbf{G}\mathbf{H}$ we can compute the global reduction as $\mathrm{d}(\mathbf{H}) = (G : H)$, while from the alignment relation $\mathbf{J}_0 = \mathbf{G}_0 \boldsymbol{\Delta}$ we can evaluate the reduction with respect to each vector. For instance, if $\mathrm{d}(\mathbf{H}) = 24$ the global reduction is of 24 times, but knowing that $\boldsymbol{\Delta}$ is, e.g., $\mathrm{diag}\,(3,2,4)$, we find that the reduction is 3 times along $\mathbf{g}_1$, 2 times along $\mathbf{g}_2$ and 4 times along $\mathbf{g}_3$. This knowledge may find applications in down–sampling and up–periodization.

Example 5.1. Consider the 2D bases

$$\mathbf{G} = \begin{bmatrix} 3 & 1 \\ 0 & 1 \end{bmatrix}, \ \mathbf{J} = \begin{bmatrix} 14 & 2 \\ 2 & 2 \end{bmatrix} \quad \to \quad \mathbf{H} = \mathbf{G}^{-1}\mathbf{J} = \begin{bmatrix} 4 & 0 \\ 2 & 2 \end{bmatrix}.$$

Since $\mathrm{d}(\mathbf{H}) = 8$ we find that the sublattice J is 8 times sparser than G. The Smith decomposition gives

$$\mathbf{H} = \begin{bmatrix} 4 & 0 \\ 2 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix} = \mathbf{E}_1\,\boldsymbol{\Delta}\,\mathbf{E}_2$$

Hence, the aligned bases are

$$\mathbf{G}_0 = \mathbf{G}\,\mathbf{E}_1 = \begin{bmatrix} 7 & 3 \\ 1 & 0 \end{bmatrix}, \qquad \mathbf{J}_0 = \mathbf{J}\,\mathbf{E}_2^{-1} = \begin{bmatrix} 14 & 12 \\ 2 & 0 \end{bmatrix}, \qquad \mathbf{J}_0 = \mathbf{G}_0\,\mathrm{diag}\,[2,4]\,.$$

Consequently: $\mathbf{j}_1 = 2\mathbf{g}_1$ and $\mathbf{j}_2 = 4\mathbf{g}_2$, as shown in Fig.5.1.

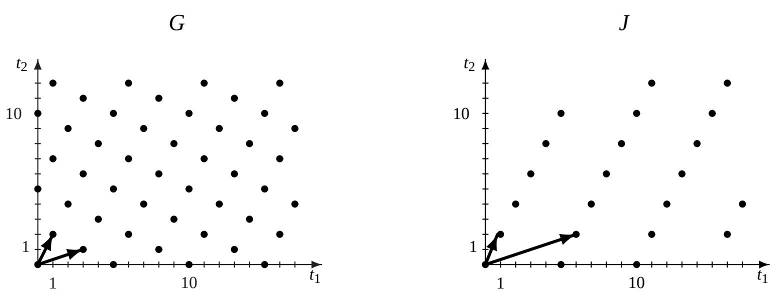


Fig. 5.1 Aligned bases of two 2D lattices

5.1.2 *The case of degenerate lattices*

If the lattice and the sublattice are reduced-dimensional the alignment of their bases can be handled using Theorem 3.1 seen in Chap.3.

5.1.3 *Alignment of the reciprocal lattices*

If J is a sublattice of G, then $G^\star$ becomes a sublattice of $J^\star$. Now, it is easy to see that, once obtained the alignment of the bases G and J, it is immediate to have the aligned bases of $J^\star$ and $G^\star$. In fact, recalling that a basis ($\mathbf{G}^\star$) of the reciprocal is given by the inverse transpose of the original bases ($\mathbf{G}^\star = (\mathbf{G}^{-1})'$), from (5.2) one gets

$$\mathbf{G}_0^\star = \mathbf{G}^\star \mathbf{E}_1^\star, \qquad \mathbf{J}_0^\star = \mathbf{J}^\star (\mathbf{E}_2^{-1})^\star$$

where $\mathbf{E}_1^\star$ and $(\mathbf{E}_2^{-1})^\star = \mathbf{E}_2'$ are still unimodular.

5.2 The Procedure "basealignment"

The procedure is simply obtained using the Smith decomposition according to Theorem 5.1.

Usage

```
{g,j,diag} = basealignment[a,h]
```

where

- `a` is the basis of the lattice,
- `h` is is an integer matrix with $\mathrm{d}(h) > 1$, which gives the basis of a sublattice as `a.h`,
- `g` and `j` are the aligned bases,
- `diag` is the diagonal matrix relating the two aligned bases.

5.3 Demos of Alignment

The demo procedures are contained in the file `demo4macro.m`

5.3.1 *The procedures* `writealignment` *and* `showalignment`

Usage

```
writealignment[a,h]
```

where

- `a` is the basis of the lattice,
- `h` is is an integer matrix with $\mathrm{d}(h) > 1$, which gives the basis of a sublattice as `a.h`,

The procedure `showalignment[a,h]` write in TEX as `writealignment` and gives also the graphical representation in the 2D and 3D cases.

Example 5.2. (*2D lattice/sublattice*) With the `Mathematica` statements

```
lab="AL112"
initdemo4[lab]
a22={{2,1},
     {0,1}}
b22={{6,2},
     {0,1}}
showalignment[a22,b22]
completedemo4
```

TEX and `Mathematica` give:

Basis alignment: run $\boxed{\textbf{AL112}}$ **(May 26, 2011)**
Given: a basis $\mathbf{G}$ of the lattice G and a reduction matrix $\mathbf{H}$ (such that $\mathbf{J} = \mathbf{G}\mathbf{H}$ is a basis of a sublattice J), the procedure `writealignment` write the upper-triangular bases $\mathbf{G}_u$ and $\mathbf{J}_u$, the Smith decomposition of $\mathbf{H}$ and the aligned bases $\mathbf{G}_0$ and $\mathbf{J}_0$.

$$\mathbf{G} = \begin{bmatrix} 2 & 5 \\ 1 & 1 \end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix} 6 & 2 \\ 0 & 1 \end{bmatrix} \qquad \mathbf{J} = \begin{bmatrix} 12 & 9 \\ 6 & 3 \end{bmatrix}$$

Upper triangular bases

$$\mathbf{G}_u = \begin{bmatrix} 3 & 2 \\ 0 & 1 \end{bmatrix} \qquad \mathbf{J}_u = \begin{bmatrix} 6 & 3 \\ 0 & 3 \end{bmatrix}$$

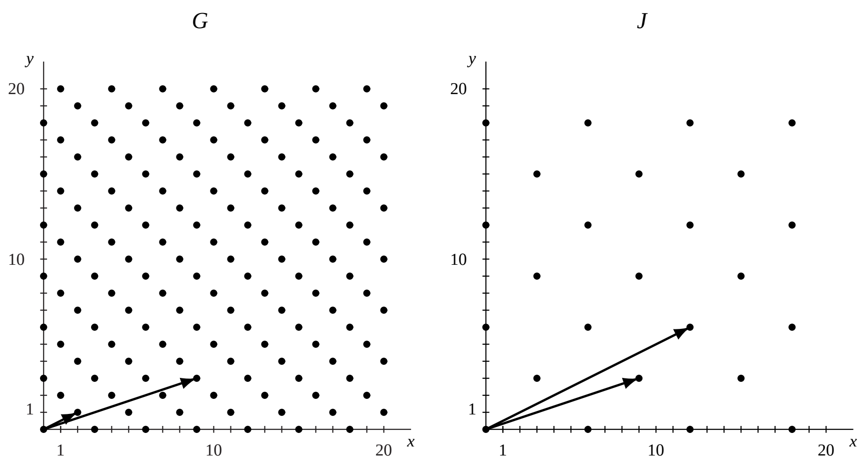
Smith diagonalization of $\mathbf{H}$:

$$\mathbf{H} = \mathbf{E}_1 \Delta\, \mathbf{E}_2 \qquad \begin{bmatrix} 6 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 6 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Aligned bases:

$$\mathbf{G}_0 = \begin{bmatrix} 9 & 2 \\ 3 & 1 \end{bmatrix} \qquad \mathbf{J}_0 = \begin{bmatrix} 9 & 12 \\ 3 & 6 \end{bmatrix} \qquad \Delta = \begin{bmatrix} 1 & 0 \\ 0 & 6 \end{bmatrix}$$

showalignment



Example 5.3. (*3D lattice/sublattice*) With the `Mathematica` statements

```
lab="AL114"
initdemo4[lab]
a33={{2,1,0},
     {0,3,0},
     {0,1,2}}
h33={{2,0,0},   (* buone da H *)
    {0,3,0},
    {0,4,1}}
showalignment[a33,h33]
completedemo4
```

TEX and `Mathematica` give:

Basis alignment: run $\boxed{\textbf{AL114}}$ **(May 26, 2011)**

Given: a basis $\mathbf{G}$ of the lattice G and a reduction matrix $\mathbf{H}$ (such that $\mathbf{J} = \mathbf{G}\mathbf{H}$ is a basis of a sublattice J), the procedure `writealignment` write the upper-triangular bases $\mathbf{G}_u$ and $\mathbf{J}_u$, the Smith decomposition of $\mathbf{H}$ and the aligned bases $\mathbf{G}_0$ and $\mathbf{J}_0$.

$$\mathbf{G} = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 3 & 0 \\ 0 & 1 & 2 \end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 4 & 1 \end{bmatrix} \qquad \mathbf{J} = \begin{bmatrix} 4 & 3 & 0 \\ 0 & 9 & 0 \\ 0 & 11 & 2 \end{bmatrix}$$

Upper triangular bases

$$\mathbf{G}_u = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 6 & 3 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{J}_u = \begin{bmatrix} 4 & 2 & 3 \\ 0 & 18 & 9 \\ 0 & 0 & 1 \end{bmatrix}$$

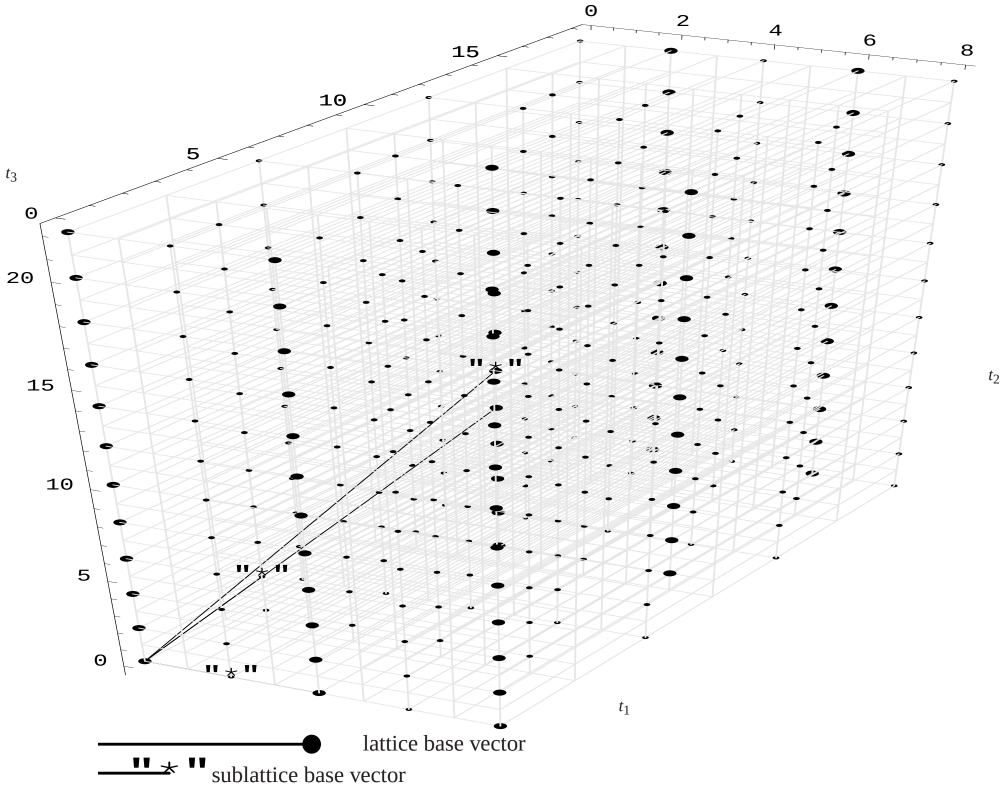
Smith diagonalization of $\mathbf{H}$:

$$\mathbf{H} = \mathbf{E}_1 \Delta\, \mathbf{E}_2 \qquad \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 4 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 1 \\ 0 & 4 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & -1 \end{bmatrix}$$

Aligned bases:

$$\mathbf{G}_0 = \begin{bmatrix} 2 & 3 & 1 \\ 0 & 9 & 3 \\ 0 & 11 & 3 \end{bmatrix} \qquad \mathbf{J}_0 = \begin{bmatrix} 4 & 3 & 3 \\ 0 & 9 & 9 \\ 0 & 11 & 9 \end{bmatrix} \qquad \Delta = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

showalignment



Example 5.4. (*degenerate lattice/sublattice*)
With the `Mathematica` statements

```
lab="AL124"
Get["demo4macro.m"]
initdemo4[lab]
ad33 = {{6,2,2},
          {12,4,2},
          {0,0,9}}
bd33 = {{2,0,1},
          {0,3,0},
          {0,0,2}}
showalignment[ad33,bd33]
completedemo4
```

TEX and `Mathematica` give:

Basis alignment: run $\boxed{\text{AL124}}$ (May 26, 2011)

Given: a basis $\mathbf{G}$ of the lattice G and a reduction matrix $\mathbf{H}$ (such that $\mathbf{J} = \mathbf{G}\mathbf{H}$ is a basis of a sublattice J), the procedure `writealignment` write the upper-triangular bases $\mathbf{G}_u$ and $\mathbf{J}_u$, the Smith decomposition of $\mathbf{H}$ and the aligned bases $\mathbf{G}_0$ and $\mathbf{J}_0$.

$$\mathbf{G} = \begin{bmatrix} 6 & 2 & 2 \\ 12 & 4 & 2 \\ 0 & 0 & 9 \end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \end{bmatrix} \qquad \mathbf{J} = \begin{bmatrix} 12 & 6 & 10 \\ 24 & 12 & 16 \\ 0 & 0 & 18 \end{bmatrix}$$

Upper triangular bases

$$\mathbf{G}_u = \begin{bmatrix} 0 & 2 & 2 \\ 0 & 4 & 2 \\ 0 & 0 & 9 \end{bmatrix} \qquad \mathbf{J}_u = \begin{bmatrix} 0 & 6 & 4 \\ 0 & 12 & 4 \\ 0 & 0 & 18 \end{bmatrix}$$

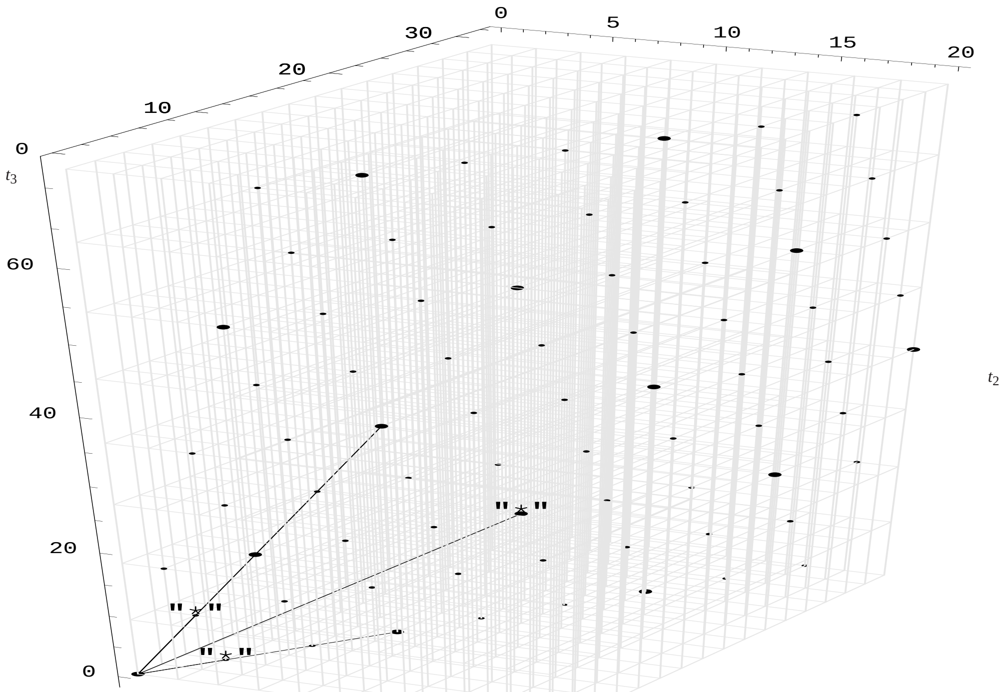
Smith diagonalization of $\mathbf{H}$:

$$\mathbf{H} = \mathbf{E}_1 \Delta\, \mathbf{E}_2 \qquad \begin{bmatrix} 2 & 0 & 1 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} 2 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

Aligned bases:

$$\mathbf{G}_0 = \begin{bmatrix} 10 & 2 & 2 \\ 16 & 4 & 2 \\ 18 & 0 & 9 \end{bmatrix} \qquad \mathbf{J}_0 = \begin{bmatrix} 10 & 6 & 8 \\ 16 & 12 & 8 \\ 18 & 0 & 36 \end{bmatrix} \qquad \Delta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

showalignment



In this case the lattice/sublattice are degenerate because `rank[ad33]=2`.

lattice base vector

"⋆" sublattice base vector

Example 5.5. (*lattice/sublattice with dimension* $m = 4$)

With the `Mathematica` statements

```
lab="AL108"
Get["demo4macro.m"]
initdemo4[lab]
a44 = {{2,0,0,1},
       {0,3,0,2},
       {0,0,2,3},
       {0,1,2,3}}
h44 = {{6,2,2,1},
       {0,4,2,2},
       {6,0,9,3},
       {1,2,2,1}}
writealignmentlarge[a44,h44]

completedemo4
```

TEX and `Mathematica` give:

Basis alignment: run $\boxed{\textbf{AL108}}$ (May 26, 2011)

Given: a basis $\mathbf{G}$ of the lattice G and a reduction matrix $\mathbf{H}$ (such that $\mathbf{J} = \mathbf{G}\mathbf{H}$ is a basis of a sub-lattice J), the procedure `writealignment` write the upper-triangular bases $\mathbf{G}_u$ and $\mathbf{J}_u$, the Smith decomposition of $\mathbf{H}$ and the aligned bases $\mathbf{G}_0$ and $\mathbf{J}_0$.

$$\mathbf{G} = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 0 & 3 & 0 & 2 \\ 0 & 0 & 2 & 3 \\ 0 & 1 & 2 & 3 \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} 6 & 2 & 2 & 1 \\ 0 & 4 & 2 & 2 \\ 6 & 0 & 9 & 3 \\ 1 & 2 & 2 & 1 \end{bmatrix} \quad \mathbf{J} = \begin{bmatrix} 13 & 6 & 6 & 3 \\ 2 & 16 & 10 & 8 \\ 15 & 6 & 24 & 9 \\ 15 & 10 & 26 & 11 \end{bmatrix}$$

Upper triangular bases

$$\mathbf{G}_u = \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 4 & 3 & 3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{J}_u = \begin{bmatrix} 10 & 2 & 6 & 9 \\ 0 & 8 & 6 & 0 \\ 0 & 0 & 6 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Smith diagonalization of $\mathbf{H}$:

$$\mathbf{H} = \mathbf{E}_1 \Delta\, \mathbf{E}_2 \quad \begin{bmatrix} 6 & 2 & 2 & 1 \\ 0 & 4 & 2 & 2 \\ 6 & 0 & 9 & 3 \\ 1 & 2 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & -2 & 1 & 0 \\ 3 & 9 & -4 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 12 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix} \begin{bmatrix} 6 & 2 & 2 & 1 \\ -60 & -6 & -5 & 0 \\ -11 & -1 & -1 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}$$

Aligned bases:

$$\mathbf{G}_0 = \begin{bmatrix} 3 & 0 & 0 & 1 \\ 8 & -6 & 3 & 2 \\ 9 & 18 & -8 & 3 \\ 11 & 16 & -7 & 3 \end{bmatrix} \quad \mathbf{J}_0 = \begin{bmatrix} 3 & 0 & 0 & 5 \\ 8 & -6 & 36 & 10 \\ 9 & 18 & -96 & 15 \\ 11 & 16 & -84 & 15 \end{bmatrix} \quad \Delta = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 12 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}$$

In this case the graphical representation is not possible.

Chapter 6
Continuous Cells

The topic of cells is considered in Chapter 3 and Chapter 16 of the *Unified Signal Theory*. This chapter deals with the implementation in `Mathematica` of cells of the type $\mathbb{R}^m$/lattice, which we call for brevity *continuous cells*. *Discrete cells* will be seen in the next chapter.

The most common example of continuous cell is given by the fundamental parallelepiped, which is identified by the bases of the lattice. Considering the multiplicity of the bases of a lattice, we may find a large variety of cells of this type. The possibility of a *centered* fundamental parallelepiped increases this class. Moreover, it is possible the construction of *orthogonal* cells (see *Unified Signal Theory*, Section 16.9).

Another form of cell that will be implemented is the Voronoi cell.

Procedures of this chapter

The basic procedures, contained in the file `lattices.m`, are:

- `cell2D`: gives the vertexes of a 2D cell related to the fundamental parallelepiped,
- `voronoi2D`: gives the list of the vertexes of a 2D Voronoi cell,
- `cell3D`: gives the faces of a 3D cell related to the fundamental parallelepiped,
- `voronoi3D`: gives the list of generating points and the list of the vertexes of each face of a 3D Voronoi cell,
- `vert2D`: gives the coordinates of a point that is equidistant from the origin and two other fixed points,
- `axisprimitive`: gives the primitive points of a lattice that belongs to the coordinate axes.

The demo procedures, contained in the file `demo5macro.m`, are:

a) `writecell2D[a,type]`

b) `writeallcell2D[a]`
c) `writecell3D[a,type,cent,vp]`
d) `writecellbasis[a,ul,vp]`
e) `writecellallbases[a]`
f) `showcell2D[a,type,factor,axes]`
g) `showallcells2D[a,region,axes]`
h) `showcell3D[a,type,cent,vp]`
i) `showvoronoi3D[a]`

6.1 Definition of Continuous Cell

From Section 3.5 of the *Unified Signal Theory* we recall the general definition of a *cell*.

Definition 6.1. Let G be an Abelian group and let C and P non–empty subsets of G. The set C is a *cell* of G *modulo* P, denoted by[1] $[G/P)$, if the shifted replicas of C. The set

$$C+p \stackrel{\Delta}{=} \{c+p \,|\, c \in C\}, \qquad p \in P \tag{6.1}$$

represents a *partition* of G, that is,

$$\begin{aligned} &(C+p)\cap(C+q)=\emptyset, \qquad p \neq q \ , \ p,q \in P \\ &\bigcup_{p\in P}(C+p)=G\,. \end{aligned} \tag{6.2}$$

The partition of group G, can be written synthetically in the form

$$\boxed{[G/P)+P=G\,.} \tag{6.3}$$

The modulus P is called the *set of repetition centers* (thinking that P is a lattice). A cell can be interpreted as follows: by shifting C over all the repetition centers, the group G is covered without superposition.

We note that:

1) If C is a cell, also every shifted–replica $C+p_0$ with $p_0 \in G$ is a cell, in particular with $p_0 \in P \subset G$. For this reason, the class (6.1) represents *a partition of the group G* into cells.
2) For a given pair G,P the cell partition is not unique.
3) If $P=\{0\}$, the unique cell is $C=G$.

From the general definition with $G=\mathbb{R}^m$ and P a lattice a $\mathbb{R}^m$ we obtain the definition of continuous cell.

---

[1] This symbol, proposed by the author, recalls that a cell of $\mathbb{R}$ modulus $\mathbb{Z}(T)$ is given by the half–open interval $[0,T)$.

Definition 6.2. Let P be a lattice of $\mathbb{R}^m$. Then C is a *continuous cell of* $\mathbb{R}^m$ *modulo* P if the sets obtained from C by shifts belonging to P:

$$C+p \stackrel{\Delta}{=} \{c+p \mid c \in C\} \quad , \qquad p \in P \tag{6.4}$$

form a *partition* of $\mathbb{R}^m$, that is,

$$\begin{aligned} &(C+p)\cap(C+q)=\emptyset \quad , \qquad p \neq q \; , \; p,q \in P \\ &\bigcup_{p\in P}(C+p)=\mathbb{R}^m \quad . \qquad \square \end{aligned} \tag{6.5}$$

A continuous cell is indicated in the form $[\mathbb{R}^m/P)$.

6.2 Fundamental parallelepiped

Let P be a lattice and $=[\mathbf{p}_1 \cdots \mathbf{p}_m]$ and let $\mathbf{P}$ be a basis of P. Then, the set

$$C=\{\alpha_1\mathbf{p}_1+\cdots+\alpha_m\mathbf{p}_m \mid 0\le\alpha_1<1,\ldots,0\le\alpha_m<1\}=\mathbf{P}\,[0,1)^m \tag{6.6}$$

is a cell $[\mathbb{R}^m/P)$.

Fig. 6.1 shows the fundamental parallelepiped in the cases of a separable lattice and of a nonseparable lattice.

The set

$$C_c=\{\alpha_1\mathbf{p}_1+\cdots+\alpha_m\mathbf{p}_m \mid -\tfrac{1}{2}\le\alpha_1<\tfrac{1}{2},\ldots,-\tfrac{1}{2}\le\alpha_m<\tfrac{1}{2}\}=\mathbf{P}\,[-\tfrac{1}{2},\tfrac{1}{2})^m$$

is a shift of the fundamental parallelepiped (6.6)

$$C_c=C_0+(-\tfrac{1}{2}\,\mathbf{p}_1,\ldots,-\tfrac{1}{2}\,\mathbf{p}_m) \tag{6.7}$$

and therefore it is still a cell $[\mathbb{R}^m/P)$.

Considering that the basis of a lattice P is not unique, we may find several fundamental parallelepipeds. We recall that if $\mathbf{P}_0$ is a basis of P, all the bases $\mathbf{P}$ are obtained as

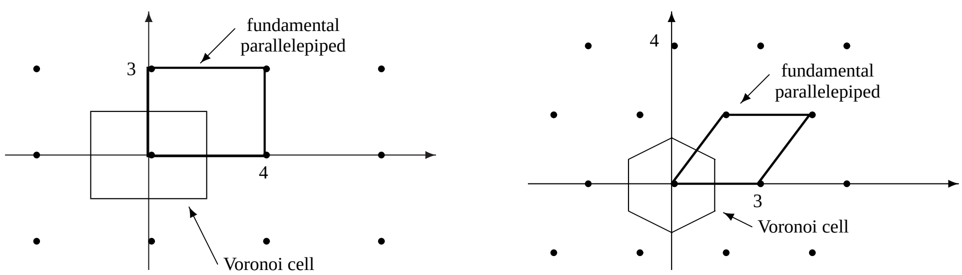


Fig. 6.1 Example of a cell $\mathbb{R}^2$ modulo $\mathbb{Z}(4,3)$ and of a cell $\mathbb{R}^2$ modulo $\mathbb{Z}_2^1(\tfrac{3}{2},2)$

$$\mathbf{P} = \mathbf{P}_0\,\mathbf{E}$$

where $\mathbf{E}$ is a unimodular matrix.

Definition 6.3. A basis $\mathbf{P}$ will be called a *cell–generating basis*. The generated cell is a fundamental parallelepiped, according to (6.6) or (6.7).

6.2.1 Orthogonal Cells

The reference lattice $G_0 = \mathbb{Z}(d_1,\dots,d_m)$ has an *orthogonal* fundamental parallelepiped $C_0 = [0,d_1)\times\cdots\times[0,d_m)$. If a lattice P is not separable it is still possible to get an orthogonal cell. The general statement is given by Theorem 16.8 of the *Unified Signal Theory*:

Theorem 6.1. *Let L be a lattice of $\mathcal{L}_m(G_0)$ and let $\mathbf{G}_0\mathbf{U}$ be a triangular basis of L. Then*

$$\mathbf{G}_0\,[0,u_{11})\times\cdots\times[0,u_{mm}) = [\mathbb{R}^m/P) \tag{6.8}$$

where u_{ii} are the diagonal entries of $\mathbf{U}$, is a cell of $\mathbb{R}^m$ modulo P.

The theorem is illustrated in Fig.6.2 in the 2D case.

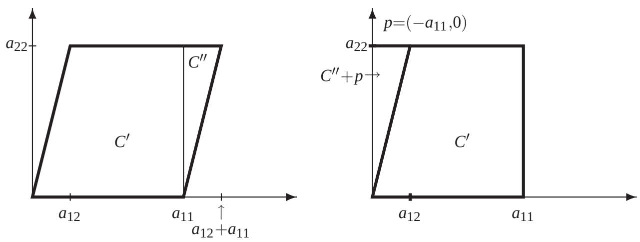


Fig. 6.2 Orthogonal cell obtained with the procedure “cut anf past”

6.3 Voronoi Cells

From Section 16.9 of the *Unified Signal Theory* we recall:

Definition 6.4. Given a lattice L of $\mathbb{R}^m$, the Voronoi cell $\mathcal{V}_m(L)$ is the subset of $\mathbb{R}^m$ given by the points that are *nearer to the origin* than any other lattice point.

To get familiarity with Voronoi cells the reader has to reconsider the following concepts of Section 16.9:

- primitive points,
- primitive axis points,
- generating points.
- inner product criterion (see eq. (16.71)).

6.4 Procedures for 2D Continuous Cells

We have implemented with `Mathematica` 9 types of 2D continuous cells, namely:

type = 1, fundamental parallelepiped,

type = 2, centered fundamental parallelepiped,

type = 3, fundamental parallelepiped related to triangU,

type = 4, centered fundamental parallelepiped related to triangU,

type = 5, fundamental parallelepiped related to triangL,

type = 6, centered fundamental parallelepiped related to triangL,

type = 7, orthogonal parallelepiped,

type = 8, centered orthogonal parallelepiped,

type = 9, Voronoi cell

In all these cases the 2D continuous cell is a polygon, which is normally a parallelogram, but in the case of Voronoi it may become an hexagon. In any case, the identification of continuous cell is given by the set **V** of its vertexes. The evaluation of vertexes is obtained according to (6.6) or (6.7), with a modification in the case of orthogonal cells; for Voronoi cells see Chapter 16 of the *Unified Signal Theory*.

The procedure is given by the following `Mathematica` code, written in `lattice.m`:

```
cell2D[a_,type_]:=Module[{aa = Transpose[a],
      parfond ={{0, 0}, {1, 0}, {1, 1}, {0, 1}},
      parcent ={{-1/2,-1/2},{1/2,-1/2},{1/2,1/2},
        {-1/2,1/2}}, vertexes},
 vertexes = Which[
    type == 1,    (* fund. parallelogram*)
                    parfond .aa ,
    type == 2,    (* centered fund. parallelogram*)
                    parcent .aa,
    type == 3,    (* triangular U.*)
                    parfond . Transpose[triangU[a]] ,
    type == 4,    (* triangolare U.*)
                    parcent . Transpose[triangU[a]] ,
    type == 5,    (* triangular L.*)
                    parfond . Transpose[triangL[a]],
    type == 6,    (* triangular L.*)
                    parcent .Transpose[ triangL[a]],
    type == 7,    (* orthogonal*)
          b = triangU[a]; b=ReplacePart[b,0,{1,2}];
                    parfond . b ,
    type == 8,    (* centered orthogonal *)
          b = triangU[a]; b=ReplacePart[b,0,{1,2}];
                    parcent . b ,
    type == 9,    (*  Voronoi *) voronoi2D[a]   ] ]
```

Usage

`vertexes = cell2D[a,type]`

where

- `a` is a basis of the 2D lattice,
- `type` is the type of cell, as listed above,
- `vertexes` is the set of vertexes.

6.4.1 *Demos of 2D Cells*

The following demo procedures (contained in the file `demo5macro.m`) illustrate the several types of 2D continuous cells:

- `writecell2D[a,type]`
- `showcell2D[a,type,region]`
- `writeallcell2D[a]`
- `showallcells2D[a,region]`

Usage of Demo Procedure "writecell2D" and "showcell2D"

`writecell2D[a,type]`

where

- `a` is a basis of the 2D lattice,
- `type` is the type of cell, as listed above.

The procedure write in TEX the vertexes of the cell.

`showcell2D[a,type,region,axes]`

where

- `a` is a basis of the 2D lattice,
- `type` is the type of cell, as listed above,
- `region` is the region of the plane containing the cell,
- `axes` is an optional argument with default value 0 (with `axes=1` one gets the axes coordinates).

The procedure writes in TEX the vertexes of the cell and also gives the graphical representation.

Example 6.1. With the `Mathematica` statements

```
Get["demo5macro.m"]
lab="CC202"
initdemo5[lab]
 a22={{0,8},
     {12,20}}
 type=9
writecell2D[a22,type]
completedemo5
```

TEX and `Mathematica` give:

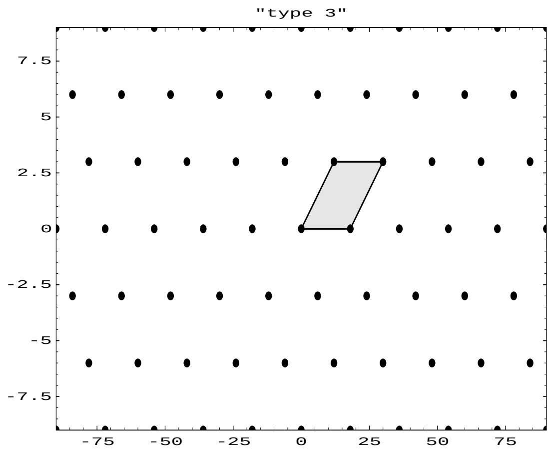
Continuous cells: run $\boxed{\textbf{CC202}}$ **(May 26, 2011)**
Demo `writecell2D`
Given **G**: lattice basis

$$\mathbf{G} = \begin{bmatrix} 0 & 6 \\ 9 & 15 \end{bmatrix}$$

the procedure writes the vertexes collected as the columns of the matrix **V**

type =3 fundamental parall. of triangU $\mathbf{V} = \begin{bmatrix} 0 & 18 & 30 & 12 \\ 0 & 0 & 3 & 3 \end{bmatrix}$

Demo `showcell2D`
factor=3



Example 6.2. With the `Mathematica` statements

```
Get["demo5macro.m"]
lab="CC205"
initdemo5[lab]
 a22={{0,8},
     {12,20}}
 type=9
showcell2D[a22,type]
completedemo5
```

TEX and `Mathematica` give:

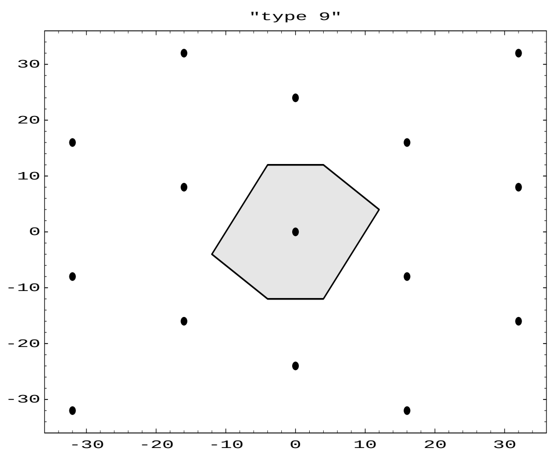
Continuous cells: run $\boxed{\textbf{CC206}}$ (May 26, 2011)
Demo `writecell2D`
Given **G**: lattice basis

$$\mathbf{G} = \begin{bmatrix} 0 & 16 \\ 24 & 40 \end{bmatrix}$$

the procedure writes the vertexes collected as the columns of the matrix **V**

type =9 Voronoi cell $\mathbf{V} = \begin{bmatrix} -12 & -4 & 4 & 12 & 4 & -4 \\ -4 & -12 & -12 & 4 & 12 & 12 \end{bmatrix}$

Demo `showcell2D`
factor=3



Example 6.3. With the `Mathematica` statements

```
Get["demo5macro.m"]
lab="CC210"
initdemo5[lab]
 a22={{0,8},
     {12,20}}
 type=9
showallcells2D[a22,type]
completedemo5
```

TEX and `Mathematica` give:

Continuous cells: run $\boxed{\textbf{CC210}}$ **(May 26, 2011)**
Given the integer matrix **G**

$$\mathbf{G} = \begin{bmatrix} 0 & 6 \\ 9 & 15 \end{bmatrix}$$

`showallcells2D` gives the generating matrix $\mathbf{G}_0$ (for **type**$\neq$ 9) and the vertexes **V** of the 9 cells generated by **G**

$\mathbf{G}_0 = \begin{bmatrix} 0 & 6 \\ 9 & 15 \end{bmatrix}$ $\mathbf{V} = \begin{bmatrix} 0 & 0 \\ 0 & 9 \end{bmatrix}$ 1 fundamental parall.

$\mathbf{G}_0 = \begin{bmatrix} 0 & 6 \\ 9 & 15 \end{bmatrix}$ $\mathbf{V} = \begin{bmatrix} -3 & -3 \\ -12 & -3 \end{bmatrix}$ 2 centered fundamental parall.

$\mathbf{G}_0 = \begin{bmatrix} 18 & 12 \\ 0 & 3 \end{bmatrix}$ $\mathbf{V} = \begin{bmatrix} 0 & 18 \\ 0 & 0 \end{bmatrix}$ 3 fundamental parall. of triangU

$\mathbf{G}_0 = \begin{bmatrix} 18 & 12 \\ 0 & 3 \end{bmatrix}$ $\mathbf{V} = \begin{bmatrix} -15 & 3 \\ -\frac{3}{2} & -\frac{3}{2} \end{bmatrix}$ 4 centered fundamental parall. of triangU

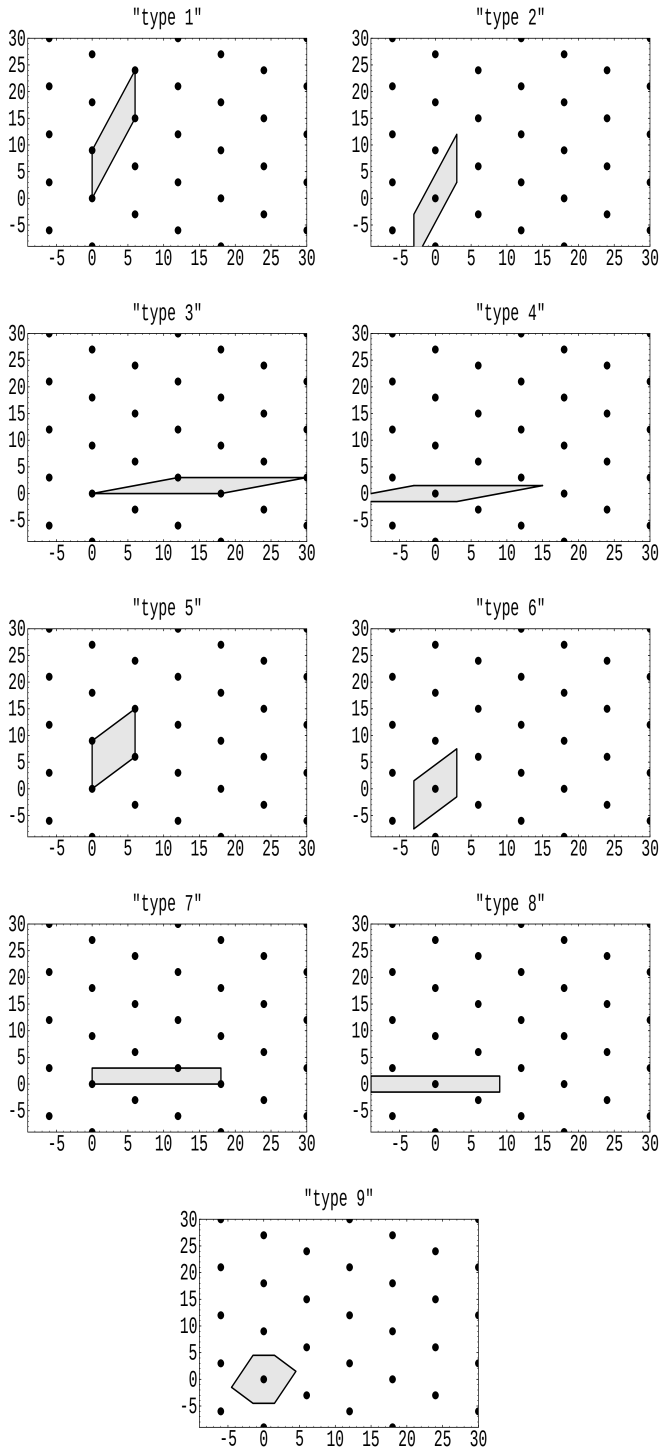
$\mathbf{G}_0 = \begin{bmatrix} 6 & 0 \\ 6 & 9 \end{bmatrix}$ $\mathbf{V} = \begin{bmatrix} 0 & 6 \\ 0 & 6 \end{bmatrix}$ 5 fundamental parall. of triangL

$\mathbf{G}_0 = \begin{bmatrix} 6 & 0 \\ 6 & 9 \end{bmatrix}$ $\mathbf{V} = \begin{bmatrix} -3 & 3 \\ -\frac{15}{2} & -\frac{3}{2} \end{bmatrix}$ 6 centered fundamental parall. of triangL

$\mathbf{G}_0 = \begin{bmatrix} 18 & 0 \\ 0 & 3 \end{bmatrix}$ $\mathbf{V} = \begin{bmatrix} 0 & 18 \\ 0 & 0 \end{bmatrix}$ 7 orthogonal parall.

$\mathbf{G}_0 = \begin{bmatrix} 18 & 0 \\ 0 & 3 \end{bmatrix}$ $\mathbf{V} = \begin{bmatrix} -9 & 9 \\ -\frac{3}{2} & -\frac{3}{2} \end{bmatrix}$ 8 centered orthogonal parall.

$\mathbf{V} = \begin{bmatrix} -\frac{9}{2} & -\frac{3}{2} \\ -\frac{3}{2} & -\frac{9}{2} \end{bmatrix}$ 9 Voronoi cell


"type 1"
"type 2"
"type 3"
"type 4"
"type 5"
"type 6"
"type 7"
"type 8"
"type 9"

6.5 Procedures for 3D Continuous Cells

We have implemented with `Mathematica` the following types of 3D continuous cells:

type = G, fundamental parallelepiped,

type = U, fundamental parallelepiped related to triangU,

type = UO, orthogonal fundamental parallelepiped related to triangU,

type = L, fundamental parallelepiped related to triangL,

type = LO, orthogonal fundamental parallelepiped related to triangL,

type = V, Voronoi cell.

In all these cases the 3D continuous cell is a polyhedron, which is normally a parallelepiped, but in the case of Voronoi cell it may become an octahedron. In any case, the identification of the continuous cell is given by the set **V** of its faces, which are 2D polygons (rectangles or hexagons) in $\mathbb{R}^3$.

The fundamental procedure is `cell3D` written in the file `lattices.m`

Usage

`{gen,faces} =cell3D[a,type,cent,vp]`

where

- `a` is a basis of the 3D lattice,
- `type` is the type of 3D cell, as listed above,
- `cent` refers to the ordinary parallelepiped with `cent=0` and to the centered parallelepiped with `cent=1`,
- `vp` is an optional argument with default $\{1,2,3\}$ which indicates the possible permutation of coordinates.
- `gen` gives the generating matrix (the generating points in the case of Voronoi),
- `faces` give the set of faces.

6.5.1 *Demos of 3D Cells: "showcell3D"*

The demo procedures (contained in the file `demo5macro.m`) illustrates the type of 3D cells.

- `writecell3D[a,type,cent,vp]`,
- `showcell3D[a,type,cent,vp]`
- `showvoronoi3D[a]`

The arguments have the meaning indicated above for the procedure `cell3D[a,type,cent,vp]`.

Example 6.4. With the `Mathematica` statements

```
Get["demo5macro.m"];
lab="CC226"
initdemo5[lab]
d33={{4,2,2},
     {4,0,1},
     {2,4,4}}
showcell3D[d33,"U0",0]
completedemo5
```

TEX and `Mathematica` give:

Continuous cells: run $\boxed{\textbf{CC226}}$ **(May 26, 2011)**

Given **G**: the lattice basis, the type (G,L,LO,U,UO,V), cent (0 or 1), optional permutation of $\{1,2,3\}$:

$$\mathbf{G} = \begin{bmatrix} 4 & 2 & 2 \\ 4 & 0 & 1 \\ 2 & 4 & 4 \end{bmatrix} \qquad \text{type=UO} \qquad \text{cent=0} \qquad \mathbf{p} = [1 \quad 2 \quad 3]$$

the procedure `writecell3D` writes the generating basis $\mathbf{G}_0$ and the faces **F**.

Generating matrix

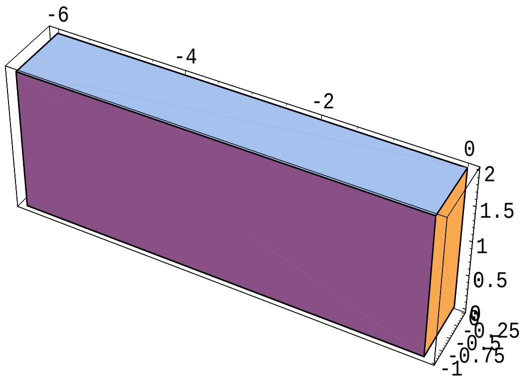
$$\mathbf{G}_0 = \begin{bmatrix} -6 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

Faces

$$F_1 = \begin{bmatrix} 0 & -6 & -6 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \quad F_2 = \begin{bmatrix} 0 & -6 & -6 \\ 0 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad F_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 2 & 2 \end{bmatrix}$$

$$F_4 = \begin{bmatrix} -6 & -6 & -6 \\ 0 & 0 & -1 \\ 0 & 2 & 2 \end{bmatrix} \quad F_5 = \begin{bmatrix} 0 & -6 & -6 \\ 0 & 0 & -1 \\ 2 & 2 & 2 \end{bmatrix} \quad F_6 = \begin{bmatrix} 0 & -6 & -6 \\ -1 & -1 & -1 \\ 0 & 0 & 2 \end{bmatrix}$$

Demo `showcell3D`



Example 6.5. With the `Mathematica` statements

```
Get["demo5macro.m"]
lab="CC224"
initdemo5[lab]
a33={{2,0,3},
    {3,0,2},
    {4,-1,7}}
showcell3D[a33,"L",0]
completedemo5
```

TEX and `Mathematica` give:

Continuous cells: run $\boxed{\text{CC224}}$ **(May 26, 2011)**

Given **G**: the lattice basis, the type (G,L,LO,U,UO,V), cent (0 or 1), optional permutation of $\{1,2,3\}$:

$$\mathbf{G} = \begin{bmatrix} 2 & 0 & 3 \\ 3 & 0 & 2 \\ 4 & -1 & 7 \end{bmatrix} \qquad \text{type=L} \qquad \text{cent=0} \qquad \mathbf{p} = [1 \quad 2 \quad 3]$$

the procedure `writecell3D` writes the generating basis $\mathbf{G}_0$ and the faces $\mathbf{F}$.

Generating matrix

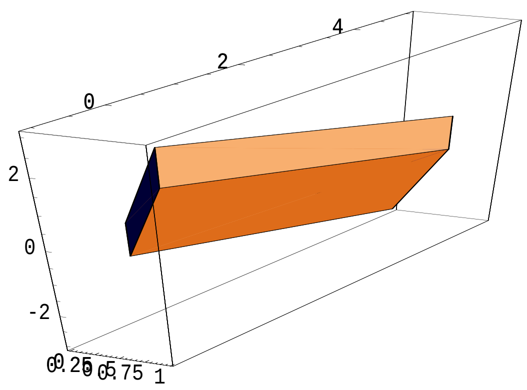
$$\mathbf{G}_0 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 5 & 0 \\ 3 & -2 & -1 \end{bmatrix}$$

Faces

$$F_1 = \begin{bmatrix} 0 & 1 & 1 \\ 0 & -1 & 4 \\ 0 & 3 & 1 \end{bmatrix} \quad F_2 = \begin{bmatrix} 0 & 1 & 1 \\ 0 & -1 & -1 \\ 0 & 3 & 2 \end{bmatrix} \quad F_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 5 \\ 0 & -1 & -3 \end{bmatrix}$$

$$F_4 = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & 4 \\ 3 & 2 & 0 \end{bmatrix} \quad F_5 = \begin{bmatrix} 0 & 1 & 1 \\ 0 & -1 & 4 \\ -1 & 2 & 0 \end{bmatrix} \quad F_6 = \begin{bmatrix} 0 & 1 & 1 \\ 5 & 4 & 4 \\ -2 & 1 & 0 \end{bmatrix}$$

Demo `showcell3D`



6.5.2 *Demos of 3D Cells: "showvoronoi3D"*

Usage

```
showvoronoi3D[a}
```

where

- `a` is a basis of the 3D lattice,

Example 6.6. With the `Mathematica` statements

```
Get["demo5macro.m"]
lab="CC220"
initdemo5[lab]
 c3={{6,0,2},
     {0,4,0},
     {0,0,7}}
showvoronoi3D[c3]
completedemo5
```

TEX and `Mathematica` give:

Continuous cells: run $\boxed{\textbf{CC220}}$ **(May 26, 2011)**

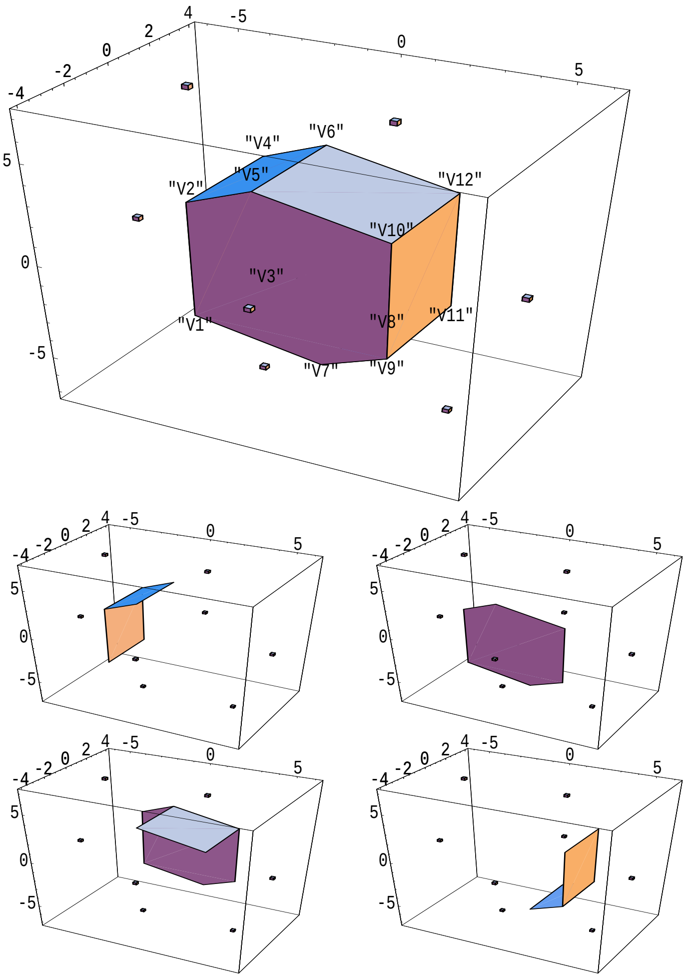
Given the basis of a 3D lattice

$$\mathbf{G} = \begin{bmatrix} 6 & 0 & 2 \\ 0 & 4 & 0 \\ 0 & 0 & 7 \end{bmatrix}$$

the procedure `showvoronoi3D` evaluate the faces of the Voronoi cell 3D and shows the cell and its faces.

---

axis primitive points $[6 \quad 4 \quad 21]$
N. of primitive points in the region46
N. of generating points 8
N. of faces 8



$$\text{face1}=\begin{bmatrix} -3 & -3 & -3 \\ -2 & 2 & 2 \\ -\frac{41}{14} & -\frac{41}{14} & \frac{41}{14} \end{bmatrix}$$

$$\text{face2}=\begin{bmatrix} -3 & -1 & -1 \\ -2 & -2 & 2 \\ \frac{41}{14} & \frac{57}{14} & \frac{57}{14} \end{bmatrix}$$

$$\text{face3}=\begin{bmatrix} -3 & -3 & 1 \\ -2 & 2 & 2 \\ -\frac{41}{14} & -\frac{41}{14} & -\frac{57}{14} \end{bmatrix}$$

$$\text{face4}=\begin{bmatrix} -3 & 1 & 3 \\ -2 & -2 & -2 \\ -\frac{41}{14} & -\frac{57}{14} & -\frac{41}{14} \end{bmatrix}$$

$$\text{face5}=\begin{bmatrix} -3 & 1 & 3 \\ 2 & 2 & 2 \\ -\frac{41}{14} & -\frac{57}{14} & -\frac{41}{14} \end{bmatrix}$$

$$\text{face6}=\begin{bmatrix} -1 & -1 & 3 \\ -2 & 2 & 2 \\ \frac{57}{14} & \frac{57}{14} & \frac{41}{14} \end{bmatrix}$$

$$\text{face7}=\begin{bmatrix} 1 & 3 & 3 \\ -2 & -2 & 2 \\ -\frac{57}{14} & -\frac{41}{14} & -\frac{41}{14} \end{bmatrix}$$

$$\text{face8}=\begin{bmatrix} 3 & 3 & 3 \\ -2 & 2 & 2 \\ -\frac{41}{14} & -\frac{41}{14} & \frac{41}{14} \end{bmatrix}$$

Example 6.7. With the Mathematica statements

```
Get["demo5macro.m"]
lab="CC228"
initdemo5[lab]
 aa33={{1,2,3},
         {3,0,2},
         {4,-1,3}}
   showvoronoi3D[aa33]
completedemo5
```

TEX and Mathematica give:

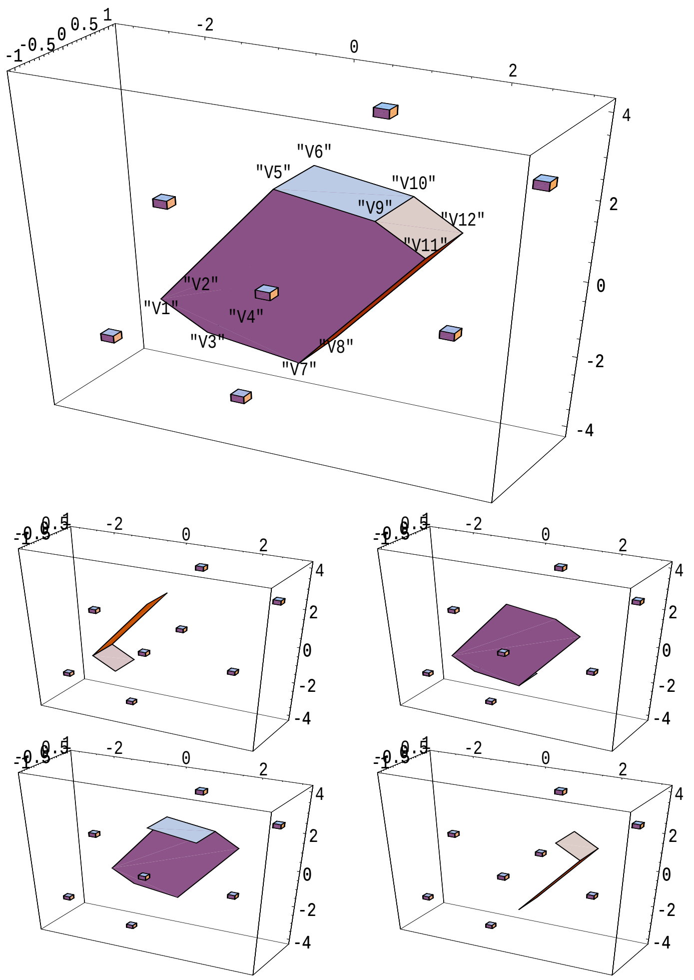
Continuous cells: run $\boxed{\textbf{CC228}}$ **(May 26, 2011)**
Given the basis of a 3D lattice

$$\mathbf{G} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 0 & 2 \\ 4 & -1 & 3 \end{bmatrix}$$

the procedure `showvoronoi3D` evaluate the faces of the Voronoi cell 3D and shows the cell and its faces.

---

axis primitive points $[9 \quad 1 \quad 9]$
N. of primitive points in the region110
N. of generating points 8
N. of faces 8



$$\text{face1}=\begin{bmatrix} -\frac{11}{6} & -\frac{7}{6} & -\frac{7}{6} \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ -\frac{7}{6} & -\frac{11}{6} & -\frac{11}{6} \end{bmatrix}$$

$$\text{face2}=\begin{bmatrix} -\frac{11}{6} & -\frac{11}{6} & -\frac{1}{6} \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ -\frac{7}{6} & -\frac{7}{6} & \frac{13}{6} \end{bmatrix}$$

$$\text{face3}=\begin{bmatrix} -\frac{7}{6} & -\frac{7}{6} & \frac{1}{6} \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ -\frac{11}{6} & -\frac{11}{6} & -\frac{13}{6} \end{bmatrix}$$

$$\text{face4}=\begin{bmatrix} -\frac{11}{6} & -\frac{7}{6} & \frac{1}{6} \\ -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{7}{6} & -\frac{11}{6} & -\frac{13}{6} \end{bmatrix}$$

$$\text{face5}=\begin{bmatrix} -\frac{11}{6} & -\frac{7}{6} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ -\frac{7}{6} & -\frac{11}{6} & -\frac{13}{6} \end{bmatrix}$$

$$\text{face6}=\begin{bmatrix} -\frac{1}{6} & -\frac{1}{6} & \frac{7}{6} \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{13}{6} & \frac{13}{6} & \frac{11}{6} \end{bmatrix}$$

$$\text{face7}=\begin{bmatrix} \frac{1}{6} & \frac{1}{6} & \frac{11}{6} \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ -\frac{13}{6} & -\frac{13}{6} & \frac{7}{6} \end{bmatrix}$$

$$\text{face8}=\begin{bmatrix} \frac{7}{6} & \frac{11}{6} & \frac{11}{6} \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \frac{11}{6} & \frac{7}{6} & \frac{7}{6} \end{bmatrix}$$

Example 6.8. With the `Mathematica` statements

```
Get["demo5macro.m"]
lab="CC230"
initdemo5[lab]
  b33={{0,0,5},
        {2,0,0},
        {0,3,0}}
   showvoronoi3D[b33]
completedemo5
```

TEX and `Mathematica` give:

Continuous cells: run $\boxed{\textbf{CC230}}$ **(May 26, 2011)**

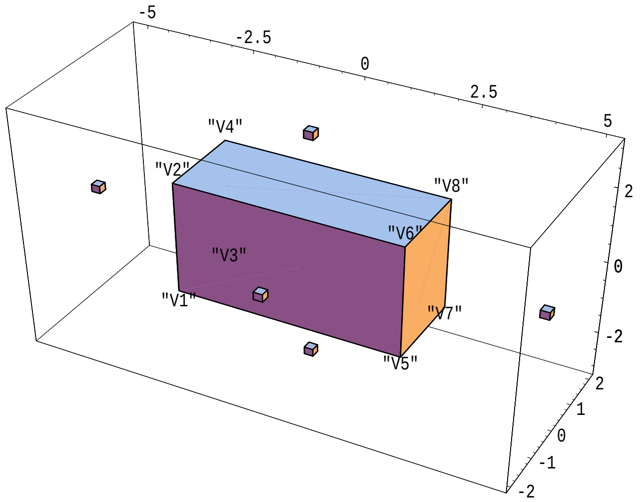
Given the basis of a 3D lattice

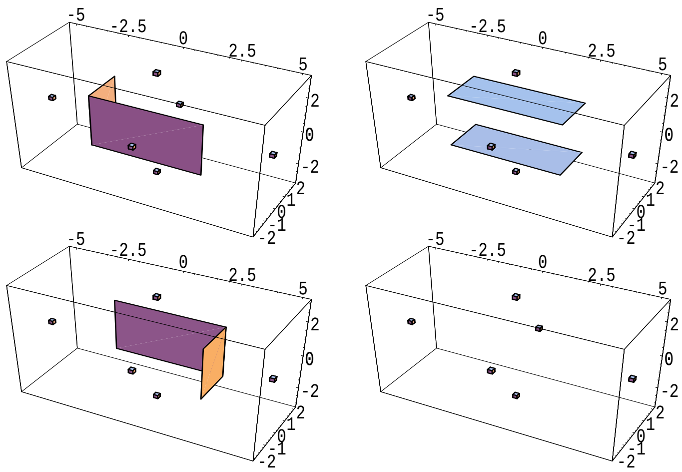
$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 5 \\ 2 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix}$$

the procedure `showvoronoi3D` evaluate the faces of the Voronoi cell 3D and shows the cell and its faces.

---

axis primitive points $[5 \quad 2 \quad 3]$
N. of primitive points in the region26
N. of generating points 6
N. of faces 6





$$\text{face1}=\begin{bmatrix} -\frac{5}{2} & -\frac{5}{2} & -\frac{5}{2} \\ -1 & 1 & 1 \\ -\frac{3}{2} & -\frac{3}{2} & \frac{3}{2} \end{bmatrix}$$

$$\text{face2}=\begin{bmatrix} -\frac{5}{2} & -\frac{5}{2} & \frac{5}{2} \\ -1 & -1 & -1 \\ -\frac{3}{2} & \frac{3}{2} & \frac{3}{2} \end{bmatrix}$$

$$\text{face3}=\begin{bmatrix} -\frac{5}{2} & -\frac{5}{2} & \frac{5}{2} \\ -1 & 1 & 1 \\ -\frac{3}{2} & -\frac{3}{2} & -\frac{3}{2} \end{bmatrix}$$

$$\text{face4}=\begin{bmatrix} -\frac{5}{2} & -\frac{5}{2} & \frac{5}{2} \\ -1 & 1 & 1 \\ \frac{3}{2} & \frac{3}{2} & \frac{3}{2} \end{bmatrix}$$

$$\text{face5}=\begin{bmatrix} -\frac{5}{2} & -\frac{5}{2} & \frac{5}{2} \\ 1 & 1 & 1 \\ -\frac{3}{2} & \frac{3}{2} & \frac{3}{2} \end{bmatrix}$$

$$\text{face6}=\begin{bmatrix} \frac{5}{2} & \frac{5}{2} & \frac{5}{2} \\ -1 & 1 & 1 \\ -\frac{3}{2} & -\frac{3}{2} & \frac{3}{2} \end{bmatrix}$$

6.6 Cells with arbitrary dimensions

For a dimension $m \geq 4$ it is possible to obtain the continuous cells of the type of fundamental parallelepiped. The demo procedure `writecellallbases` gives the generating matrices obtained from the triangular representations with an arbitrary permutation of the coordinates.

Demo procedure "writecellallbases"

Usage

`writecellallbases[a]`

where

- `a` is a basis of a lattice with an arbitrary dimension.

The demo evaluates the triangular forms $\mathbf{G}_L$ and $\mathbf{G}_U$ and all the possible cell generating bases obtained from $\mathbf{G}_L$ and $\mathbf{G}_U$ with permutation of the coordinates.

Example 6.9. (fundamental parallelepipeds 4D)
With the `Mathematica` statements

```
Get["demo5macro.m"]
lab="CC240"
initdemo5[lab]
(************************************)
   a44={{3,3,2,0},
        {1,7,5,1},
        {0,3,0,7},
        {1,1,3,2}}
   writecellallbases[a44]
(************************************)
completedemo5
```

TEXwrites 24 cells of type L and 24 cells of type U

Continuous cells: run $\boxed{\text{CC240}}$ (May 26, 2011)
Demo `writecellallbases`
Given a lattice basis

$$\mathbf{G} = \begin{bmatrix} 3 & 3 & 2 & 0 \\ 1 & 7 & 5 & 1 \\ 0 & 3 & 0 & 7 \\ 1 & 1 & 3 & 2 \end{bmatrix}$$

`writecellallbases` evaluates the triangular forms $\mathbf{G}_L$: triangular L and $\mathbf{G}_U$: triangular U and all the possible cell generating bases obtained from $\mathbf{G}_L$ and $\mathbf{G}_U$ with permutation of the coordinates

$$\mathbf{G}_L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 2 & 7 & 13 & 0 \\ 16 & 2 & 22 & 27 \end{bmatrix} \qquad \mathbf{G}_U = \begin{bmatrix} 351 & 172 & 124 & 182 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [1 \quad 2 \quad 3 \quad 4] \qquad \mathbf{G}_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 2 & 7 & 13 & 0 \\ 16 & 2 & 22 & 27 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [1 \quad 2 \quad 4 \quad 3] \qquad \mathbf{G}_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 184 & 293 & 208 & 351 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [1 \quad 3 \quad 2 \quad 4] \qquad \mathbf{G}_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 9 & 2 & 13 & 0 \\ 0 & 1 & 0 & 0 \\ 5 & 9 & 7 & 27 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [1 \quad 3 \quad 4 \quad 2] \qquad \mathbf{G}_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 100 & 236 & 52 & 351 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [1 \quad 4 \quad 2 \quad 3] \qquad \mathbf{G}_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 184 & 208 & 293 & 351 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [1 \quad 4 \quad 3 \quad 2] \qquad \mathbf{G}_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 100 & 52 & 236 & 351 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [2 \quad 1 \quad 3 \quad 4] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 7 & 2 & 13 & 0 \\ 2 & 16 & 22 & 27 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [2 \quad 1 \quad 4 \quad 3] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 293 & 184 & 208 & 351 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [2 \quad 3 \quad 1 \quad 4] \qquad \mathbf{G}_p = \begin{bmatrix} 3 & 7 & 13 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 9 & 2 & 27 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [2 \quad 3 \quad 4 \quad 1] \qquad \mathbf{G}_p = \begin{bmatrix} 172 & 124 & 182 & 351 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [2 \quad 4 \quad 1 \quad 3] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 293 & 208 & 184 & 351 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [2 \quad 4 \quad 3 \quad 1] \qquad \mathbf{G}_p = \begin{bmatrix} 172 & 182 & 124 & 351 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [3 \quad 1 \quad 2 \quad 4] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 2 & 9 & 13 & 0 \\ 1 & 0 & 0 & 0 \\ 9 & 5 & 7 & 27 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [3 \quad 1 \quad 4 \quad 2] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 236 & 100 & 52 & 351 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [3 \quad 2 \quad 1 \quad 4] \qquad \mathbf{G}_p = \begin{bmatrix} 7 & 3 & 13 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 9 & 1 & 2 & 27 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [3 \quad 2 \quad 4 \quad 1] \qquad \mathbf{G}_p = \begin{bmatrix} 124 & 172 & 182 & 351 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [3 \quad 4 \quad 1 \quad 2] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 236 & 52 & 100 & 351 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [3 \quad 4 \quad 2 \quad 1] \qquad \mathbf{G}_p = \begin{bmatrix} 124 & 182 & 172 & 351 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [4 \quad 1 \quad 2 \quad 3] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 208 & 184 & 293 & 351 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [4 \quad 1 \quad 3 \quad 2] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 52 & 100 & 236 & 351 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [4 \quad 2 \quad 1 \quad 3] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 208 & 293 & 184 & 351 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [4 \quad 2 \quad 3 \quad 1] \qquad \mathbf{G}_p = \begin{bmatrix} 182 & 172 & 124 & 351 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [4 \quad 3 \quad 1 \quad 2] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 52 & 236 & 100 & 351 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{L} \qquad \mathbf{p} = [4 \quad 3 \quad 2 \quad 1] \qquad \mathbf{G}_p = \begin{bmatrix} 182 & 124 & 172 & 351 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{U} \qquad \mathbf{p} = [1 \quad 2 \quad 3 \quad 4] \qquad \mathbf{G}_p = \begin{bmatrix} 351 & 172 & 124 & 182 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

U $\quad \mathbf{p} = [1 \quad 2 \quad 4 \quad 3] \quad \mathbf{G}_p = \begin{bmatrix} 351 & 172 & 182 & 124 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

U $\quad \mathbf{p} = [1 \quad 3 \quad 2 \quad 4] \quad \mathbf{G}_p = \begin{bmatrix} 351 & 124 & 172 & 182 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

U $\quad \mathbf{p} = [1 \quad 3 \quad 4 \quad 2] \quad \mathbf{G}_p = \begin{bmatrix} 351 & 124 & 182 & 172 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

U $\quad \mathbf{p} = [1 \quad 4 \quad 2 \quad 3] \quad \mathbf{G}_p = \begin{bmatrix} 351 & 182 & 172 & 124 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$

U $\quad \mathbf{p} = [1 \quad 4 \quad 3 \quad 2] \quad \mathbf{G}_p = \begin{bmatrix} 351 & 182 & 124 & 172 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$

U $\quad \mathbf{p} = [2 \quad 1 \quad 3 \quad 4] \quad \mathbf{G}_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 351 & 100 & 236 & 52 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

U $\quad \mathbf{p} = [2 \quad 1 \quad 4 \quad 3] \quad \mathbf{G}_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 351 & 100 & 52 & 236 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

U $\quad \mathbf{p} = [2 \quad 3 \quad 1 \quad 4] \quad \mathbf{G}_p = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 351 & 236 & 100 & 52 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

U $\quad \mathbf{p} = [2 \quad 3 \quad 4 \quad 1] \quad \mathbf{G}_p = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 351 & 236 & 52 & 100 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

$$\text{U} \qquad \mathbf{p} = [2 \quad 4 \quad 1 \quad 3] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 351 & 52 & 100 & 236 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\text{U} \qquad \mathbf{p} = [2 \quad 4 \quad 3 \quad 1] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 351 & 52 & 236 & 100 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\text{U} \qquad \mathbf{p} = [3 \quad 1 \quad 2 \quad 4] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 351 & 184 & 293 & 208 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{U} \qquad \mathbf{p} = [3 \quad 1 \quad 4 \quad 2] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 351 & 184 & 208 & 293 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\text{U} \qquad \mathbf{p} = [3 \quad 2 \quad 1 \quad 4] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 351 & 293 & 184 & 208 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{U} \qquad \mathbf{p} = [3 \quad 2 \quad 4 \quad 1] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 351 & 293 & 208 & 184 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\text{U} \qquad \mathbf{p} = [3 \quad 4 \quad 1 \quad 2] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 351 & 208 & 184 & 293 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\text{U} \qquad \mathbf{p} = [3 \quad 4 \quad 2 \quad 1] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 351 & 208 & 293 & 184 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\text{U} \qquad \mathbf{p} = [4 \quad 1 \quad 2 \quad 3] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 13 & 3 & 7 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 27 & 2 & 1 & 9 \end{bmatrix}$$

$$\text{U} \qquad \mathbf{p} = [4 \quad 1 \quad 3 \quad 2] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 13 & 7 & 3 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 27 & 2 & 9 & 1 \end{bmatrix}$$

$$\text{U} \qquad \mathbf{p} = [4 \quad 2 \quad 1 \quad 3] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 13 & 9 & 2 \\ 0 & 0 & 0 & 1 \\ 27 & 7 & 5 & 9 \end{bmatrix}$$

$$\text{U} \qquad \mathbf{p} = [4 \quad 2 \quad 3 \quad 1] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 13 & 2 & 9 \\ 0 & 0 & 1 & 0 \\ 27 & 7 & 9 & 5 \end{bmatrix}$$

$$\text{U} \qquad \mathbf{p} = [4 \quad 3 \quad 1 \quad 2] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 13 & 2 & 7 \\ 27 & 22 & 16 & 2 \end{bmatrix}$$

$$\text{U} \qquad \mathbf{p} = [4 \quad 3 \quad 2 \quad 1] \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 13 & 7 & 2 \\ 27 & 22 & 2 & 16 \end{bmatrix}$$

Chapter 7
Discrete Cells

This chapter deals with the `Mathematica` implementation of cells of the type $[G/P)$, where G is a lattice and P is a sublattice, called for brevity, *discrete cells*.

The identification of discrete cells, which consist of a finite number of points, is easy (see *Unified Signal Theory*, Chapter 16): once determined the continuous cell $C_0 = [\mathbb{R}^m/P)$, the discrete cell is simply given by the intersection $C = C_0 \cap G$. The only non trivial problem is the attribution to a discrete cell the points of the boundary of the continuous cell.

Procedures of the chapter

The specification of a discrete cell is given by a finite set of points belonging to a polygon of $\mathbb{R}^m$. To this end we have formulated two procedures:

- `polygonpoints`: gives the list of the points of a 2D lattice belonging to a given polygon;
- `polyhedronpoints`: gives the list of the points of a 3D lattice belonging to a given polyhedron.
- `discretecell2D` and `discretecell3D`: give the vertexes of the continuous cell and the points of the discrete cell.

7.1 Definition of Discrete Cell

From the general definition of cell given in Chapter 3 of the *Unified Signal Theory* (see Definition 6.1 in this manual) with $G = \mathbb{R}^m$ and P a lattice of $\mathbb{R}^m$, one gets the definition of a discrete cell:

Definition 7.1. Let G be a lattice of $\mathbb{R}^m$ and P a sublattice with full dimension. Then C is a *cell of G modulo P* if the sets obtained from C by shifts belonging to P:

$$C + p \stackrel{\Delta}{=} \{c + p \,|\, c \in C\}, \qquad p \in P \tag{7.1}$$

represents a *partition* of G, that is,

$$\begin{aligned} &(C+p)\cap(C+q)=\emptyset\,, \qquad p\neq q\;\;,\;\; p,q\in P \\ &\bigcup_{p\in P}(C+p)=G\,. \end{aligned} \tag{7.2}$$

Discrete cells have a finite cardinality given by

$$(G:P)\stackrel{\Delta}{=}\frac{\mathrm{d}(P)}{\mathrm{d}(G)}=\frac{\mu(G)}{\mu(P)}$$

which is called *index of P in G*; $\mathrm{d}(\cdot)$ and $\mu(\cdot)$ denote the determinant and the density of the lattice, respectively.

7.2 General Identification Procedure

The identification of a discrete cell C is obtained from the one of a continuous cell C_0 (see Chapter 16 of *Unified Signal Theory*) as

$$C=C_0\cap G\,.$$

In words, C is given by the points of C_0 belonging to the lattice G.

For instance, if $G=\mathbb{Z}_2^1(d_1,d_2)$ and $P=\mathbb{Z}_4^1(3d_1,3d_2)$, to get a cell $C_0=[G/P)$ we first evaluate the fundamental parallelepiped C_0 of P, considering that a basis of $\mathbb{Z}_4^1(3d_1,3d_2)$ is

$$\mathbf{P}=\begin{bmatrix}3d_1 & 8d_1\\ 3d_2 & 0\end{bmatrix}\,.$$

In such a way we get that the cell $C_0=[\mathbb{R}^2/P)$ is given by the parallelogram of Fig. 7.1. Then, we have to assign correctly the boundary points belonging to C_0; in particular, we have to include in C_0 the points of the edges OA and OB with exclusion of the vertexes A and B. Then we get the desired cell by intersection. The number of points of the cell C is $\mathrm{d}(P)/\mathrm{d}(G)=24d_1/(2d_1)=12$.

7.2.1 Discrete cells of orthogonal type

The $\mathbb{R}^m$/lattice cells of the orthogonal type identified in the previous chapter, through restriction (intersection) allow us to identify lattice/sublattice cells of the orthogonal type.

Let G and P be a lattice/sublattice pair of the class $\mathcal{L}(G_0)$. Having written for P a triangular base, from the diagonal elements $e_i d_i$, we find that

$$C_0=[0,e_1d_1)\times\cdots\times[0,e_md_m)$$

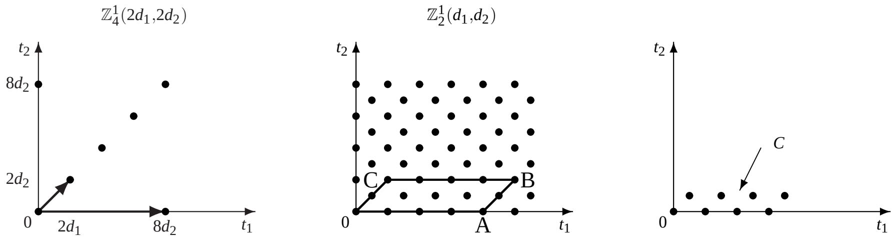


Fig. 7.1 Evaluation of a discrete cell $C = G/P$ with $G = \mathbb{Z}_2^1(d_1,d_2)$ and $P = \mathbb{Z}_4^1(2d_1,2d_2)$: C is given as the intersection of the fundamnetal parallepiped of the sublattice $\mathbb{Z}_4^1(2d_1,2d_2)$ (parallelogram 0ABC) with the lattice $\mathbb{Z}_2^1(d_1,d_2)$

is a cell $[\mathbb{R}^m/P)$. Then, by applying the rule on restriction we find that

$$C_0 \cap G = \{[0, e_1 d_1) \times \cdots \times [0, e_m d_m)\} \cap G$$

is a cell $[G/P)$.

Therefore the cell is formed by the points of the basis lattice G, bounded by the orthogonal parallelepiped C_0, that is, with coordinates $\mathbf{t} = (t_1, \ldots, t_m)$ that verify the conditions

$$0 \le t_i < e_i d_i \,.$$

The number of points of the cell is obviously given by the product $e_1 \cdots e_m$.

Fig. 7.2 shows four examples of cells on bidimensional lattices: in a), the basis lattice and the modulus are separable lattices; in b), the basis lattice is still separable, whilst the modulus is quincunx; in c) the basis lattice is quincunx, whilst the modulus is separable, and finally in d), both the basis lattice and and the modulus are quincunx. Notice for example that in the third case it results $d(G) = 2$, $d(P) = 12 \cdot 8 = 96$ and therefore $(G : P) = 48$, and in fact the cell indicated is composed by 48 points.

7.2.2 Discrete cells from aligned bases

A discrete cell, $[G/S)$ can be obtained in the following way. We search the *aligned* bases of G and P (see Chap. 5) following the steps:

1) compute the matrix $\mathbf{M} = \mathbf{G}^{-1}\mathbf{P}$, which is a matrix of integers,
2) decompose $\mathbf{M}$ through Smith's diagonalization

$$\mathbf{M} = \mathbf{E}\mathbf{D}\mathbf{F}\,, \qquad \mathbf{D} = \operatorname{diag}[d_1, \ldots, d_m]$$

   with $\mathbf{E}$ and $\mathbf{F}$ unimodular
3) the aligned bases are obtained as

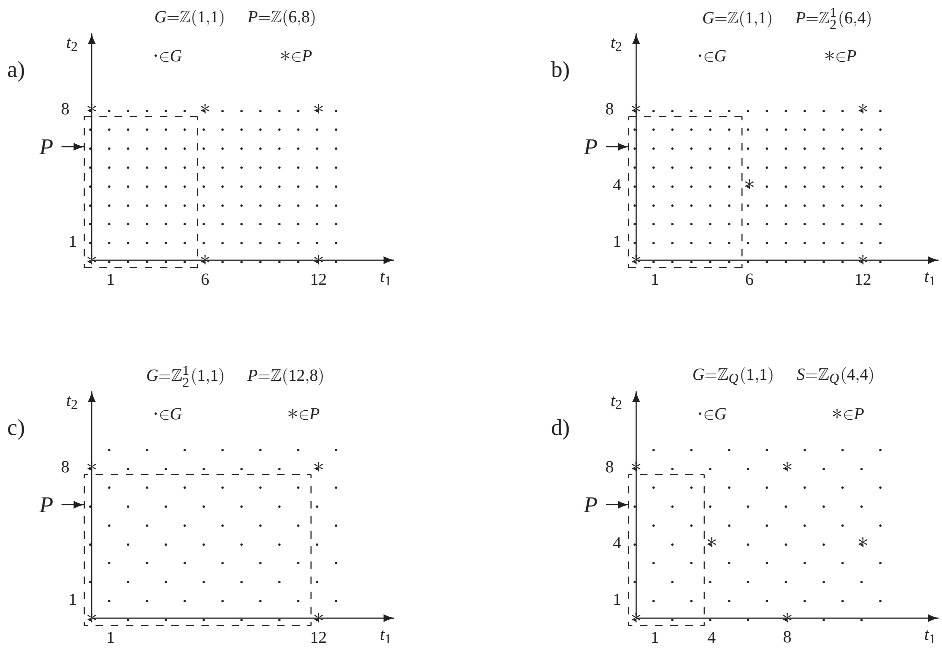


Fig. 7.2 Examplesof 2D discrete cells. The cells consist of the points contaned in the rectaungular frames

$$\mathbf{G}_0 = \mathbf{G}\mathbf{E}, \qquad \mathbf{S}_0 = \mathbf{P}\mathbf{F}^{-1}, \qquad \mathbf{S}_0 = \mathbf{G}_0\mathbf{D}$$

4) letting $\mathbf{G}_0 = [\mathbf{j}_1 \cdots \mathbf{j}_m]$ it results that

$$C = \{n_1\mathbf{j}_1 + \cdots + n_m\mathbf{j}_m | n_1 \in \mathbb{Z}_{d_1}, \ldots, n_m \in \mathbb{Z}_{d_m}\}$$

is a cell $[G/P)$, where $\mathbb{Z}_d = \{0, 1, \ldots, d-1\}$.

7.3 Description of the algorithms

Finding the cells $[G/P)$, constituted by a finite number of points

$$[G/P) = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\}, \qquad N = \mathrm{d}(P)/\mathrm{d}(G)$$

simply consists in giving a list of these points. For some kinds of cells, and precisely for the cells of the orthogonal type and for the cells from aligned bases, this list is immediately found, not only for 2D and 3D cells, but also for cells of higher dimension.

For the other types of cells obtained through *restriction* of continuous cells, we need to identify the points of a lattice belonging to a polygon in the 2D case, and to a polyhedron in the 3D case.

7.3.1 The procedure "polygonpoints"

This procedure provides the list of the points of the 2D lattice belonging to the given polygon.

Usage

```
new = polygonpoints[a,pol]
```

where

- `a` is the basis matrix of the lattice,
- `pol` is the list of the vertexes of the polygon
- `new` gives the list of the points of the lattice belonging to the polygon

Methodology. We determine the rectangular region $\mathcal{R}$ of the plane (t_1, t_2) within which we limit the computation of the lattice points: we compute the minimum and maximum values of the coordinate t_1 of the vertexes of the polygon. We do the same for the coordinate t_2. Letting $t_{1_{min}}$, $t_{1_{max}}$, $t_{2_{min}}$ and $t_{2_{max}}$ be the four values so calculated, we obtain

$$\mathcal{R} = [t_{1_{min}}, t_{1_{max}}] \times [t_{2_{min}}, t_{2_{max}}]$$

We use the procedure "latticepoints2D" to get the lattice points belonging to the rectangular region $\mathcal{R}$. Now we must choose, among such points, those belonging to the polygon. For each of them we check whether, for each side of the polygon, it belongs to the semi-plane containing the polygon.

If A, B, and C are three consecutive vertexes of the polygon and P the generic point to be checked, let **a**, **b**, **c** and **p** be the column vectors of their coordinates. The line identified by the points A and B divides the plane into two semi-planes. The points C and P belong to the same semi-plane if the determinants of the matrices

$$\mathbf{m}_1 = [\,(\mathbf{b}-\mathbf{a}) \quad (\mathbf{c}-\mathbf{a})\,] \quad \text{and} \quad \mathbf{m}_2 = [\,(\mathbf{b}-\mathbf{a}) \quad (\mathbf{p}-\mathbf{a})\,]$$

have the same sign. Briefly, we can say that if

1) $\det[\,\mathbf{m}_1 \cdot \mathbf{m}_2\,] > 0\,,$ C and P belong to the same semi-plane,

2) $\det[\,\mathbf{m}_1 \cdot \mathbf{m}_2\,] = 0\,,$ C or P belong to the line,

3) $\det[\,\mathbf{m}_1 \cdot \mathbf{m}_2\,] < 0\,,$ C and P do not belong to the same semi-plane.

7.3.2 The procedure "polyhedronpoints"

This procedure gives the list of the 3D lattice points belonging to the given polyhedron. For simplicity, we assume that the polyhedron contains the origin **O**.

Usage

```
{p,new} = polyhedronpoints[a,zacce]
```

where

- `a` is an integer matrix;
- `zacce` is the (three-dimensional) matrix having as elements the edges of the polyhedron. Each edge is specified by the list of its vertexes
- `p` is the list of the lattice points of the rectangular region containing the polyhedron
- `new` gives the list of the lattice points belonging to the polyhedron.

Methodology. The procedure mimics the operations done for procedure "polygonpoints". We determine, through the procedure "latticepoints3D", the rectangular region $\mathcal{R}$ within which to limit the computation of the lattice points. For each point we check whether it belongs to the semi-space containing the origin. If A, B, and C are three vertexes of the polygon side and P the generic point to be checked, let **a**, **b**, **c** and **p** be the column vectors of their coordinates, and π the plane identified by the side. We build the matrices

$$\mathbf{m}_1 = [\,(\mathbf{c}-\mathbf{a}) \quad (\mathbf{b}-\mathbf{a}) \quad (\mathbf{O}-\mathbf{a})\,] \quad \text{and} \quad \mathbf{m}_2 = [\,(\mathbf{c}-\mathbf{a}) \quad (\mathbf{b}-\mathbf{a}) \quad (\mathbf{p}-\mathbf{a})\,]$$

We obtain that if

1) $\det[\mathbf{m}_1 \cdot \mathbf{m}_2] > 0\,,$ P belongs to the same semi-space as the origin,

2) $\det[\mathbf{m}_1 \cdot \mathbf{m}_2] = 0\,,$ C or the origin sit on the plane π,

3) $\det[\mathbf{m}_1 \cdot \mathbf{m}_2] < 0\,,$ C does not belong to the same semi-space as the origin.

7.3.3 The procedure "discretecell2D"

Usage

```
{vertexes,pp} = discretecell2D[a,h,type]
{vertexes,pp} = discretecell3D[a,h,type]
```

where

- `a` is a lattice basis;
- `h` is a reduction matrix of the sublattice (a.h gives a basis of the sublattice)
- `vertexes` gives the vertexes of the continuous cell
- `type` is the type of cell, as listed below,
- `pp` gives the points of the discrete cell.

For 2D cells the types are:

type = 1, fundamental parallelepiped,
type = 2, centered fundamental parallelepiped,
type = 3, fundamental parallelepiped related to triangU,
type = 4, centered fundamental parallelepiped related to triangU,
type = 5, fundamental parallelepiped related to triangL,
type = 6, centered fundamental parallelepiped related to triangL,
type = 7, orthogonal parallelepiped,
type = 8, centered orthogonal parallelepiped,
type = 9, Voronoi cell

For 3D cells the types are

type = G, fundamental parallelepiped,
type = U, fundamental parallelepiped related to triangU,
type = UO, orthogonal fundamental parallelepiped related to triangU,
type = L, fundamental parallelepiped related to triangL,
type = LO, orthogonal fundamental parallelepiped related to triangL,

7.4 Demo for Discrete Cells

Usage of "showdiscretecell2D"

`showdiscretecell2D[a,h,type]`

where

- `a` is a basis of the 2D lattice,
- `h` is the reduction matrix of the sublattice (a.h gives a basis of the sublattice)
- `type` is the type of cell, as listed above.

Example 7.1. **discrete parallelepiped**

With the `Mathematica` statements

```
Get["demo6macro.m"]
lab="CD202"
initdemo6[lab]
aa22 = 2{{2, 5}, {0, 4}};
bb22 = {{12,2}, {0, 7}};
type=3
showdiscretecell2D[aa22,bb22,type]
completedemo6
```

TeX and `Mathematica` give:

Discrete cells: run $\boxed{\textbf{CD202}}$ **(May 26, 2011)**
Demo `showdiscretecell2D`
Given the lattice basis **G** and the reduction matrix **H** of the sublattice

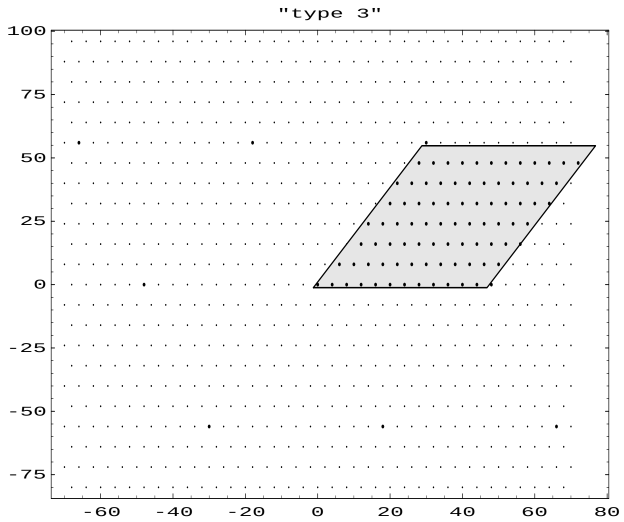
$$\mathbf{A} = \begin{bmatrix} 4 & 10 \\ 0 & 8 \end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix} 12 & 2 \\ 0 & 7 \end{bmatrix}$$

the procedure writes the vertexes collected as the columns of the matrix **V**

type =3 fundamental parall. of triangU

$$\mathbf{V} = \begin{bmatrix} 0 & 4 & 6 & 2 \\ 0 & 0 & 8 & 8 \end{bmatrix}$$

discrete cell cardinality=84



Example 7.2. **discrete Voronoi cell 2D**
With the `Mathematica` statements

```
Get["demo6macro.m"]
lab="CD204"
initdemo6[lab]
aa22 = 2{{2, 5}, {0, 4}};
bb22 = {{12,2}, {0, 7}};
type=9
showdiscretecell2D[aa22,bb22,type]
completedemo6
```

TEX and `Mathematica` give:

Discrete cells: run $\boxed{\textbf{CD204}}$ **(May 26, 2011)**
Demo `showdiscretecell2D`
Given the lattice basis **G** and the reduction matrix **H** of the sublattice

$$\mathbf{A} = \begin{bmatrix} 4 & 10 \\ 0 & 8 \end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix} 12 & 2 \\ 0 & 7 \end{bmatrix}$$

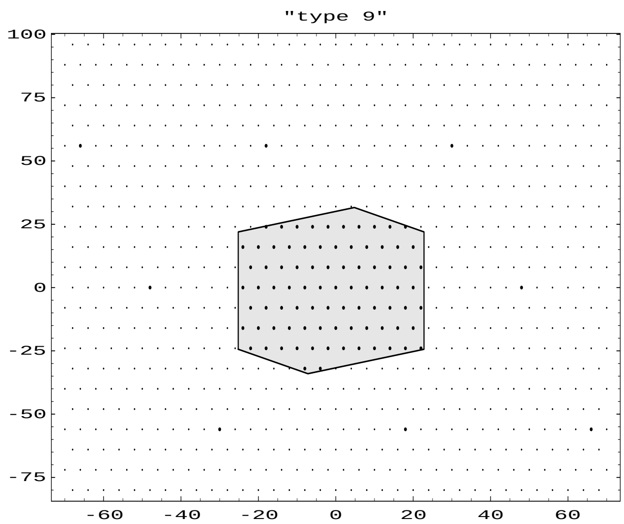
the procedure writes the vertexes collected as the columns of the matrix **V**

---

type =9 Voronoi cell

$$\mathbf{V} = \begin{bmatrix} -2 & 0 & 2 & 2 & 0 & -2 \\ 15-(--)4 & 17-(--)4 & 15-(--)4 & 15--4 & 17--4 & 15--4 \end{bmatrix}$$

discrete cell cardinality=84



Usage of "showdiscretevoooronoicell3D"

`showdiscretevoooronoicell3D[a,h,type]`

where

- `a` is a basis of the 2D lattice,
- `h` is the reduction matrix of the sublattice (a.h gives a basis of the sublattice)
- `type` is the type of cell, as listed above.

Example 7.3. **discrete Voronoi cell 3D**

With the `Mathematica` statements

```
Get["demo6macro.m"]
lab="CD214"
initdemo6[lab]
  a33={{1,0,0},
        {0,1,0},
        {0,0,1}};
   b33={{6,0,2},
        {0,4,0},
        {0,0,7}};
showdiscretevoronoicell3D[a33,b33]
completedemo6
```

TEX and `Mathematica` give:

Discrete cells: run $\boxed{\text{CD214}}$ (May 26, 2011)
Demo `showdiscretevoronoicell3D`
Given the lattice basis **G** and the reduction matrix **H** of the sublattice

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix} 6 & 0 & 2 \\ 0 & 4 & 0 \\ 0 & 0 & 7 \end{bmatrix}$$

the procedure writes the faces of the cell

---

number of primitive points= 192
number of generating points= 8
number of faces= 8

$$\mathbf{F}_1 = \begin{bmatrix} -3 & -3 & -3 \\ -2 & 2 & 2 \\ -\frac{41}{14} & -\frac{41}{14} & \frac{41}{14} \end{bmatrix}$$

$$\mathbf{F}_2 = \begin{bmatrix} -3 & -1 & -1 \\ -2 & -2 & 2 \\ \frac{41}{14} & \frac{57}{14} & \frac{57}{14} \end{bmatrix}$$

$$\mathbf{F}_3 = \begin{bmatrix} -3 & -3 & 1 \\ -2 & 2 & 2 \\ -\frac{41}{14} & -\frac{41}{14} & -\frac{57}{14} \end{bmatrix}$$

$$\mathbf{F}_4 = \begin{bmatrix} -3 & 1 & 3 \\ -2 & -2 & -2 \\ -\frac{41}{14} & -\frac{57}{14} & -\frac{41}{14} \end{bmatrix}$$

$$\mathbf{F}_5 = \begin{bmatrix} -3 & 1 & 3 \\ 2 & 2 & 2 \\ -\frac{41}{14} & -\frac{57}{14} & -\frac{41}{14} \end{bmatrix}$$

$$\mathbf{F}_6 = \begin{bmatrix} -1 & -1 & 3 \\ -2 & 2 & 2 \\ \frac{57}{14} & \frac{57}{14} & \frac{41}{14} \end{bmatrix}$$

$$\mathbf{F}_7 = \begin{bmatrix} 1 & 3 & 3 \\ -2 & -2 & 2 \\ -\frac{57}{14} & -\frac{41}{14} & -\frac{41}{14} \end{bmatrix}$$

$$\mathbf{F}_8 = \begin{bmatrix} 3 & 3 & 3 \\ -2 & 2 & 2 \\ -\frac{41}{14} & -\frac{41}{14} & \frac{41}{14} \end{bmatrix}$$