The ddphonism package*

Celia Rubio Madrigal†

August 10, 2019

Abstract

This is a music-related package which is focused on notation from the Twelve-Tone System, also called Dodecaphonism. It provides LaTeX algorithms that produce typical T.T.S. notation based off a musical series, or row sequence, of variable length.

Keywords

twelve tone system, dodecaphonism, music, mathematics, matrix, row, series, diagram, clock diagram, notation, algorithm, schoenberg, contemporary music, 20th century

Contents

1 Introduction

There are hundreds of music tools and software online which are able to produce different music notations. However, I have never seen a LaTeX tool that can do the same. This package is not only about notation, but it also calculates mathematically how this notation should work.

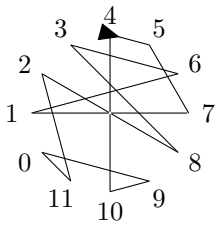
It is said that a twelve-tone matrix is the only thing a twelve-tone composer should need, because it provides the whole serial spectrum with which they may work. I wanted LaTeX users to be able to generate them automatically.

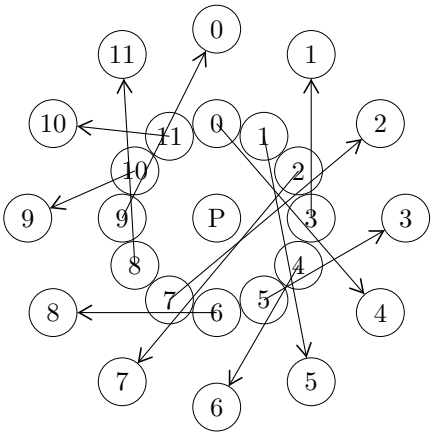
But I also think that a twelve-tone matrix is not enough, that there exist several other notations with which they may understand their series and its potential. These are the diagrams that can be obtained with this package.

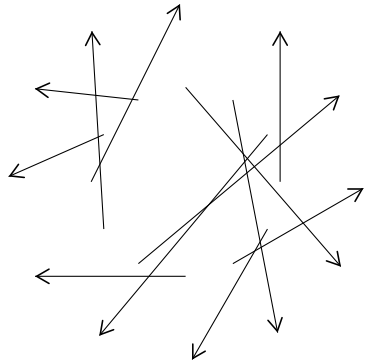
*This document corresponds to ddphonism v0.1, dated 2019/08/10.

†Email: celrubio@ucm.es

0	1	2	3	4
4	0	1	2	3
3	4	0	1	2
2	3	4	0	1
1	2	3	4	0







$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 & 0 \end{pmatrix}$$

2 Using the **ddphonism** package

These are the commands provided by ddphonism:

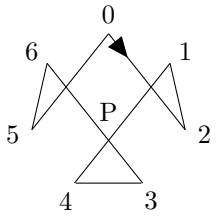
`\dmatrix` produces a twelve-tone matrix of arbitrary length, as shown in this website. Its parameter is the row sequence. For example, `\dmatrix{0,2,1,4,3,6,5}` produces the matrix

0	2	1	4	3	6	5
5	0	6	2	1	4	3
6	1	0	3	2	5	4
3	5	4	0	6	2	1
4	6	5	1	0	3	2
1	3	2	5	4	0	6
2	4	3	6	5	1	0

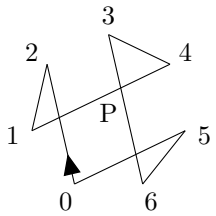
The command automatically reduces the input and output modulus the length of the input series. Thus, the input `\dmatrix{7,9,8,11,10,13,12}` produces the same output as before.

`\ddiagram` produces a twelve tone clock diagram of arbitrary length, as shown in this website. Its parameters are the row sequence and its name. An ad-

ditional optional parameter tells which number must be up north. The default value is the first number. For example, `\ddiagram{0,2,1,4,3,6,5}{P}` produces the diagram

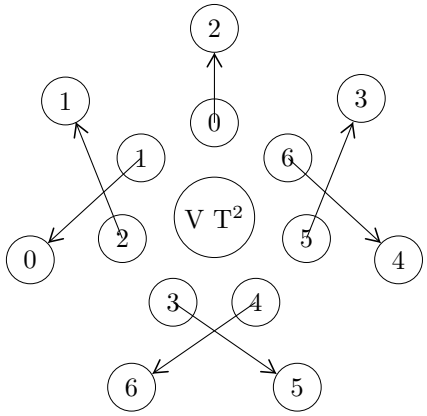


and `\ddiagram[3]{0,1,2,3,4}{P}` produces the diagram



The command also reduces the input and output modulus the length of the input series.

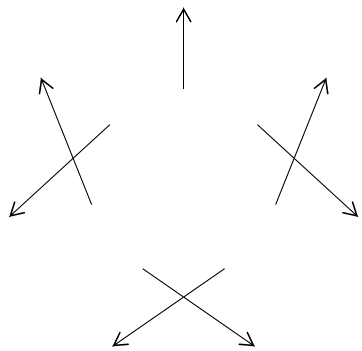
`\ddihedral` produces a dihedral representation of a series of arbitrary length. Its parameters are the original row sequence and, optionally, the transformations it undergoes: `t,s,c,v` applied in that order. For example, `\tikz{\ddihedral[t=2,v=1]{0,2,1,4,3,6,5}}` produces the diagram



It needs to be inside a `tikz` environment, as opposed to the previous ones, which are separate `tikz` environments on their own. This is because rotation or translation lines are sometimes included in this diagram, and sometimes it needs scalings.

The command also reduces the input and output modulus the length of the input series.

`\darrows` produces the arrows from the `\ddihedral` diagram. For example, `\tikz{\darrows{0,2,1,4,3,6,5}}` produces the arrows



`\drow` produces a twelve-tone row sequence as a permutation in its matrix form. For example, `\drow{0,2,1,4,3,6,5}` produces the row

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

3 The package code

```
% ddphonism
%
% (c) Celia Rubio Madrigal
%
%% This program can be redistributed and/or modified under the terms
%% of the LaTeX Project Public License Distributed from CTAN archives
%% in directory macros/latex/base/lppl.txt.

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{ddphonism}
[2019/08/10 v0.1 LaTeX package for twelve−tone matrices, clock diagrams et al.]

\RequirePackage{ifthen}
\RequirePackage{xparse}
\RequirePackage{tikz}
\RequirePackage{xstring}
\RequirePackage{pgfkeys}


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Matrices

\ usetikzlibrary {matrix}

\ExplSyntaxOn
\DeclareExpandableDocumentCommand{\Evaluation}{m}{\int_eval:n {#1}}
\ExplSyntaxOff

\newcounter{Dfirst}
```

```
\newcounter{Dsize}

\newcommand{\DsizeHead}[1]{
    \setcounter{Dsize}{0}
    \setcounter{Dfirst}{-1}

    \foreach \n in {#1}{
        \ifnum\theDfirst=-1
        \setcounter{Dfirst}{\n}
        \fi
        \stepcounter{Dsize}
    }
}

\newcommand{\Modulo}[1]{
    \ifnum#1>\Evaluation{\theDsize-1}
        \Evaluation{#1-\theDsize}
    \else\ifnum#1<0
        \Evaluation{#1+\theDsize}
    \else #1
    \fi\fi
}

\newcommand{\dmatrix}[1]{
    \DsizeHead{#1}
    \begin{tikzpicture}
        \foreach [count=\nj] \j in {#1} {
            \foreach [count=\ni] \i in {#1} {
                \draw node at (\ni-0.5,-\nj/2+0.25) {
                    \Modulo{\Evaluation{\i-\j+\theDfirst}}
                };
            }
        }
    \end{tikzpicture}
}


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Diagrams

\ usetikzlibrary {shapes,arrows,decorations.markings,shapes.misc}

\ tikzstyle ddiagramArrow=[decoration=
        {markings,mark=at position 0.25 with
            {\arrow[scale=1.25,>=triangle 45]{>}}},
    postaction={decorate}]

\newcounter{anterior}
\newcounter{var}
\newcommand{\ddiagram}[3][-1]{
    \DsizeHead{#2}
    \def\theDDfirst{\Evaluation{\Modulo{\theDfirst}}}

    \ifnum #1=-1
    \setcounter{var}{\theDDfirst}
    \else
    \setcounter{var}{#1}
```

```
    \fi

    \begin{tikzpicture}[rotate=360*\thevar/\theDsize,
        minimum height=0pt,inner sep=0pt,outer sep=0pt,scale=0.65]
    \foreach \x in {0,...,\Evaluation{\theDsize-1}} {
        \node at (90-360*\x/\theDsize:2) {\x};
        \node (\x) at (90-360*\x/\theDsize:1.6) {};
    };

    \setcounter{anterior}{-1}
    \foreach \xD in {#2}{
        \def\x{\Evaluation{\Modulo{\xD}}}
        \ifnum \theanterior=\theDDfirst
        \draw [style=ddiagramArrow] (\theanterior) -- (\x);
        \else \ifnum \theanterior=-1
        \else
        \draw (\theanterior) -- (\x);
        \fi\fi
        \setcounter{anterior}{\x}
    };
    \draw (\theanterior) -- (\theDDfirst);

    \node at (0,0) [circle, fill=white] {#3};
    \end{tikzpicture}
}


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dihedral diagrams

\tikzstyle ddihedralArrow=[decoration=
        {markings,mark=at position 1 with {\arrow[scale=1.5,>=angle 60]{>}}},
    postaction={decorate}]

\pgfkeys{
    /ddihedral/.is family, /ddihedral,
    default/.style = {t = 0, c = 0, s = 0, v = 0},
    t/.estore in = \ddihedralT,
    c/.estore in = \ddihedralC,
    s/.estore in = \ddihedralS,
    v/.estore in = \ddihedralV,
}

\newcommand{\darrows}[1]{
    \DsizeHead{#1}
    \draw foreach \x in {0,...,\Evaluation{\theDsize-1}} {
        (90-360*\x/\theDsize:2.5) node[circle] (\x) {}
    };
    \foreach \xD [count=\y] in {#1} {
        \def\x{\Evaluation{\Modulo{\xD}}}
        \draw [style=ddihedralArrow] (90-360*\Evaluation{\y-1}/\theDsize:1.25) -- (\x);
    };
}

\newcommand\ddihedral[2][]{
    \pgfkeys{/ddihedral, default, #1}

```

```
    \tikzset{inner sep=0,minimum height=18pt}

    \DsizeHead{#2}

    \draw foreach \x in {0,...,\Evaluation{\theDsize-1}} {
        (\Evaluation{(90+\ddihedralT*360/\theDsize)+(2*\ddihedralS-1)*\x*360/\theDsize}:2.5)
        node[very thin, circle ,draw] (\x) {\x}
    };

    \draw foreach \x in {0,...,\Evaluation{\theDsize-1}} {
        (\Evaluation{(90-\ddihedralC*360/\theDsize)+(2*\ddihedralV-1)*\x*360/\theDsize}:1.25)
        node[very thin, circle ,draw] {\x}
    };

    \darrows{#2}

    \node at (0,0) [very thin,draw,circle,  fill =white] {\
        \ifnum\ddihedralV=0
        \ifnum\ddihedralC=0
        \ifnum\ddihedralS=0
        \ifnum\ddihedralT=0
        P
        \fi\fi\fi
        \else V \fi
        \ifnum\ddihedralC=0
        \else C$^{\ddihedralC}$ \fi
        \ifnum\ddihedralS=0
        \else S \fi
        \ifnum\ddihedralT=0
        \else T$^{\ddihedralT}$ \fi
    \ };
}


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rows

\newcount\fooo
\long\def\addto#1#2{\expandafter\def\expandafter#1\expandafter{#1#2}}
\newcounter{myDDcntr}

\newcommand{\drow}[1]{
    \DsizeHead{#1}
    \def\tabledata{} \fooo=\theDsize
    \setcounter{myDDcntr}{0}
    \loop
    \addto\tabledata{\themyDDcntr\stepcounter{myDDcntr} &}
    \advance \fooo -1
    \ifnum \fooo>1
    \repeat
    \addto\tabledata{\themyDDcntr \\}
    \ensuremath{
        \left(\begin{array}{*{\theDsize}c}
            \tabledata
            \StrSubstitute{#1}{,}{&}\\
        \end{array}\right)
    }
```

```
}

\endinput

%% End of file 'ddphonism.sty'.
```