The \texttt{ddphonism} package\footnote{This document corresponds to \texttt{ddphonism} v0.2, dated 2019/09/01.}

Celia Rubio Madrigal\footnote{Email: \url{celrubio@ucm.es}}

September 1, 2019

Abstract

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

Keywords

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

Contents

1 Introduction 1

2 Using the \texttt{ddphonism} package 2

3 The package code 6

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 their potential. These are the diagrams that can be obtained with this package:
2 Using the \texttt{ddphonism} package

These are the commands provided by \texttt{ddphonism}. The main parameter in every command is the row sequence.

\begin{verbatim}
\texttt{dmatrix} produces a twelve-tone matrix of arbitrary length, as shown in this website. For example, \texttt{dmatrix\{0,2,1,4,3,6,5\}} produces the matrix
\begin{verbatim}
\begin{verbatim}
\begin{verbatim}
| 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 |
\end{verbatim}
\end{verbatim}
\end{verbatim}
\end{verbatim}
\end{verbatim}
\end{verbatim}

\texttt{sep} scales the matrix.
\texttt{vsep} scales the matrix vertically.
\texttt{hsep} scales the matrix horizontally.
\texttt{lines} draws lines between rows and columns.
\texttt{outside lines} only draws the outside lines.
inside lines only draws the inside lines.

vlines only draws the vertical lines.

hlines only draws the horizontal lines.

\dmatrix[lines, sep=0.75]{0,2,1,4,3,6,5} produces the matrix

\[
\begin{array}{cccccccc}
0 & 2 & 1 & 4 & 3 & 6 & 5 \\
5 & 0 & 6 & 2 & 1 & 4 & 3 \\
6 & 1 & 0 & 3 & 2 & 5 & 4 \\
3 & 5 & 4 & 0 & 6 & 2 & 1 \\
4 & 6 & 5 & 1 & 0 & 3 & 2 \\
1 & 3 & 2 & 5 & 4 & 0 & 6 \\
2 & 4 & 3 & 6 & 5 & 1 & 0
\end{array}
\]

no tikz deletes the tikz environment and lets the user write it instead.

\ddiagram produces a twelve tone clock diagram of arbitrary length, as shown in [this website]. For example, \ddiagram{0,2,1,4,3,6,5} produces the diagram

\[
\begin{tikzpicture}
\end{tikzpicture}
\]

name writes a name at the center of the diagram.

up lets the user choose which number is up north. The default value is the first number in the row.

arrow shift lets the user choose where the arrow should fall on the line. The values range from 0 to 10. The default value is 2.5.

\ddiagram[name=P, up=5, arrow shift=5]{0,2,1,4,3,6,5} produces the diagram

\[
\begin{tikzpicture}
\end{tikzpicture}
\]

no numbers deletes the numbers around the diagram.

no arrow deletes the arrow inside the diagram.
\ddiagram[no numbers, no arrow]{0,2,1,4,3,6,5} produces the diagram

\begin{tikzpicture}
\ddiagram[no tikz]{0,2,1,4,3,6,5}
\end{tikzpicture}

produces the same diagram as \ddiagram{0,2,1,4,3,6,5}.

\ddihedral produces a dihedral representation of a series of arbitrary length. For example, \ddihedral{0,2,1,4,3,6,5} produces the diagram

\begin{tikzpicture}
\ddihedral{0,2,1,4,3,6,5}
\end{tikzpicture}

The transformations are applied in that exact order.

`t` applies the transformation *transposition* to the diagram.

`s` applies the transformation *inversion* to the diagram.

`c` applies the transformation *cyclic shift* to the diagram.

`v` applies the transformation *retrograde* to the diagram.

The transformations are applied in that exact order.
\ddihedral[s=1, c=4]{0,2,1,4,3,6,5} produces the diagram

\begin{tikzpicture}
\end{tikzpicture}

no italics removes the italics from the diagram name.
new t renames the transformation transposition.
new s renames the transformation inversion.
new c renames the transformation cyclic shift.
new v renames the transformation retrograde.

\ddihedral[no italics, new v=R, v=1]{0,2,1,4,3,6,5} produces the diagram

\begin{tikzpicture}
\end{tikzpicture}

no tikz deletes the tikz environment and lets the user write it instead.
It is recommended that the user passes the option \texttt{ddihedral} to the environment:

\begin{verbatim}
\begin{tikzpicture}[ddihedral]
\ddihedral[no tikz]{0,2,1,4,3,6,5}
\end{tikzpicture}
\end{verbatim}

produces the same diagram as \texttt{\ddihedral{0,2,1,4,3,6,5}}.
The package code

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{ddphonism}
[2019/09/01 v0.2 Dodecaphonic diagrams: twelve–tone matrices, clock diagrams, etc.]
\RequirePackage{etoolbox}
\RequirePackage{xparse}
\RequirePackage{tikz}
\RequirePackage{xstring}
\RequirePackage{pgfkeys}
\documentclass{article}
\usepackage{tikz}
\usetikzlibrary{matrix}
\ExplSyntaxOn
\DeclareExpandableDocumentCommand{\Evaluation}{m}{\int_eval:n {#1}}
\ExplSyntaxOff

\newcounter{Dsize}
\newcommand{\DsizeMake}[1]{\setcounter{Dsize}{0}\foreach n in {#1}\stepcounter{Dsize}}
% Only with numbers.
\newcounter{Dfirst}
\newcommand{\DheadMake}[1]{\setcounter{Dfirst}{-1}\foreach n in {#1}\ifnum theDfirst=-1\setcounter{Dfirst}{n}\fi}
% Only when DsizeMake is already done.
\newcounter{Dmod}
\newcommand{\Modulo}[1]{\setcounter{Dmod}{#1}\loop\ifnum theDmod>\Evaluation{theDsize-1}\setcounter{Dmod}{\Evaluation{theDmod-theDsize}}\repeat\ifnum theDmod<0\setcounter{Dmod}{\Evaluation{theDmod+theDsize}}\repeat\theDmod}

\newif\ifdmatrixLines\newif\ifdmatrixOutside\newif\ifdmatrixInside\newif\ifdmatrixV\newif\ifdmatrixH\newif\ifdmatrixTikz
\pgfkeys{/dmatrix/.is family, /dmatrix, default/.style =}{lines = false, outside lines = false, inside lines = false, sep = 1, vsep = 1}
\ifdfmatrixLines
  \DLOH \DLOV \DLIH \DLIV
\fi
\ifdfmatrixOutside
  \DLOH \DLOV
\fi
\ifdfmatrixInside
  \DLIH \DLIV
\fi
\ifdfmatrixH
  \DLOH \DLIH
\fi
\ifdfmatrixV
  \DLOV \DLIV
\fi
\fi
\ifdfmatrixTikz \else %
\end{tikzpicture} % \
\fi %

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Diagrams
% %
\usetikzlibrary {shapes,arrows, decorations.markings,shapes.misc}
\tikzstyle {ddiagram} = [minimum height=0pt, inner sep=0pt, outer sep=0pt, scale=0.65]
\newif\ifddiagramTikz
\newif\ifddiagramNoNum
\newif\ifddiagramNoArr
\pgfkeys{
  /ddiagram/.is family,
  /ddiagram, default/.style =
  { name = \empty%
    , up = \empty%
    , no tikz = false
    , no numbers = false
    , no arrow = false
    , xshift = 0
    , yshift = 0
    , arrow shift = 2.5
  }
  , no tikz/.is if =ddiagramTikz
  , no numbers/.is if =ddiagramNoNum
  , no arrow/.is if =ddiagramNoArr
  , name/.estore in = \ddiagramName
  , up/.estore in = \ddiagramUp
  , xshift/.estore in = \ddiagramX
  , yshift/.estore in = \ddiagramY
  , arrow shift/.estore in = \ddiagramArrS
}
\newcounter{Dprev}
\newcommand\Dvar{}
\newcommand\ddiagram[2]{%
\DsizeMake{#2}%
\DheadMake{#2}%
\pgfkeys{/ddiagram, default, #1}%
\ifdefequal{\ddiagramUp}{empty}%
\ifdefequal{\ddiagramName}{empty}%
\ifnum\theDprev=\theDfirst%
\draw [xshift=\ddiagramX,yshift=\ddiagramY] (\theDprev) -- (\theDfirst);%
\fi
\foreach \x in {0,..., Evaluation{\theDsize - 1}}{%
\node [xshift = \ddiagramX,yshift=\ddiagramY] at (90 - 360 * \x/\theDsize:2) {\x};%
\node [xshift = \ddiagramX,yshift=\ddiagramY] (\x) at (90 - 360 * \x/\theDsize:1.6);%
\setcounter{Dprev}{-1}%
\foreach \x in {#2}{%
\ifndef \theDprev=\theDfirst%
\draw [xshift=\ddiagramX,yshift=\ddiagramY] (\theDprev) -- (\x);%
\else
\draw [xshift=\ddiagramX,yshift=\ddiagramY, decoration= {markings,mark=at position 0.999 with \ddiagramArrS with
\arrow[ scale=1.25,>={triangle 45}{>}},
postaction={decorate}]
(\theDprev) -- (\x);%
\fi
\elseifnum \theDprev=-1 \elseifnum \draw [xshift=\ddiagramX,yshift=\ddiagramY] (\theDprev) -- (\x);%
\fi
\fi
\setcounter{Dprev}{\x}%
\draw [xshift=\ddiagramX,yshift=\ddiagramY] (\theDprev) -- (\theDfirst);%
\ifndef \ddiagramName{empty}%
\ifndef \ddiagramUp{empty}%
\draw [xshift=\ddiagramX,yshift=\ddiagramY] at (0,0) [circle,fill=white] {\ddiagramName};%
\fi
\fi
}%
%
% Dihedral diagrams
\tikzstyle ddihedralArrow=[decoration={
markings,mark=at position 1 with \arrow[ scale=1.5,>=angle 60]{>}}];
\tikzstyle{ddihedral}=[inner sep=0,minimum height=18pt]
\newif\iftddihedralTikz
\newif\iftddihedralItalics
\pgfkeys{/ddihedral/.is family, /ddihedral, default/.style =
{ t = 0, c = 0, s = 0, v = 0
 , no tikz=false
 , new t = T, new c = C, new s = S, new v = V
 , no italics = false
},
no tikz/.is if=tddihedralTikz,
t/.estore in =\ddihedralT,
c/.estore in =\ddihedralC,
s/.estore in =\ddihedralS,
v/.estore in =\ddihedralV,
no italics/.is if=tddihedralItalics ,
new t/.estore in =\ddihedralNewT,
new c/.estore in =\ddihedralNewC,
new s/.estore in =\ddihedralNewS,
new v/.estore in =\ddihedralNewV,
}
\newif\ifdarrowsTikz
\pgfkeys{/darrows/.is family, /darrows, default/.style =
{ no tikz=false }
,no tikz/.is if=darrowsTikz,
}
\newcommand{\darrows}[2]{%
\DsizeMake{\#2}%
\pgfkeys{/darrows, default , #1}%
\ifdarrowsTikz\else%
\begin{tikzpicture}%
\fi%
\draw foreach \x in {0,...,\Evaluation{\theDsizel-1}} {
\Evaluation{90-360*\x/\theDsizel:2.5} node[circle] (\x) \};%
\foreach \x [count=\y] in {\#2} {
\draw [style=dihedralArrow] (\Evaluation{90-360*\y-1}/\theDsizel:1.25) -- (\x);%
\};%
\ifdarrowsTikz\else%
\end{tikzpicture}%
\fi%
}
\newcommand{\dihedral}[2]{%
\DsizeMake{\#2}%
\pgfkeys{/dihedral , default , #1}%
\iftddihedralTikz \else%
\begin{tikzpicture}{}\dihedral%
\fi%
\ensuremath{\left( \begin{array}{c} \#2 \\ \end{array} \right) }