# The ddphonism package*

Celia Rubio Madrigal†

May 13, 2025

**Abstract**

This music-related package focuses on notation from the Twelve-Tone System, also called Dodecaphonism. It provides LaTeX algorithms to generate common dodecaphonic diagrams based off a musical series, or row sequence, of arbitrary length.[1]

**Keywords**

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

## Contents



*This document corresponds to ddphonism v0.3, dated 2025/05/13.

†Email: rubiomadrigalcelia (at) gmail (dot) com

[1] The code is also hosted at `https://github.com/celrm/ddphonism`.

# 1 Introduction

There are hundreds of music tools and software available online that can produce various types of music notation. However, I have not yet seen a LaTeX tool that does the same for twelve-tone music. This package is not only about notation, but it also performs the mathematical calculations behind how the notation should work.

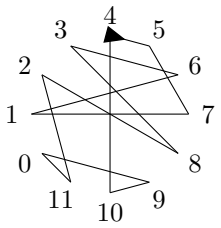
It is said that a twelve-tone matrix is the only thing a twelve-tone composer needs, because it provides the full serial spectrum they can work with. I wanted LaTeX users to be able to generate these automatically.

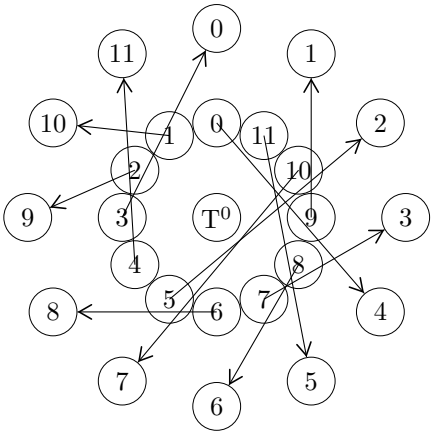
But I also think that a twelve-tone matrix is not enough, that there are other notations that can help composers better understand their series and their potential. These are the kinds of diagrams this package can produce:

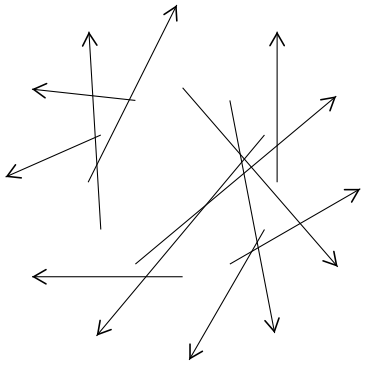
## 1.1 Examples

```
\dmatrix{4,3,2,1,0}     \ddiagram[arrow shift = 4]{4,5,7,1,6,3,8,2,11,0,9,10}

\ddihedral{4,5,7,1,6,3,8,2,11,0,9,10}     \darrows{4,5,7,1,6,3,8,2,11,0,9,10}

\drow{4,3,2,1,0}
```

| | | | | |
|---|---|---|---|---|
| 4 | 3 | 2 | 1 | 0 |
| 0 | 4 | 3 | 2 | 1 |
| 1 | 0 | 4 | 3 | 2 |
| 2 | 1 | 0 | 4 | 3 |
| 3 | 2 | 1 | 0 | 4 |



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

## 2 Using the ddphonism package

These are the commands provided by ddphonism, along with their options and examples of usage. The main parameter for every command is the main (original) row sequence. It is a comma-separated list of integers. An optional parameter can be added to include other Tikz options, such as scale or rotate.

### 2.1 \dmatrix

`\dmatrix` produces a twelve-tone matrix of arbitrary length (as seen here).

Options:

| | |
|---|---|
| sep | adjusts general spacing. |
| vsep | adjusts vertical spacing. |
| hsep | adjusts horizontal spacing. |
| lines | draws all lines surrounding rows and columns. |
| outside lines | draws only the outer box. |
| inside lines | draws only the internal grid. |
| vlines | draws only vertical lines. |
| hlines | draws only horizontal lines. |
| no tikz | disables automatic TikZ environment, letting the user type it: |

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

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

Example: `\dmatrix{0,2,1,4,3,6,5}`

| | | | | | | |
|---|---|---|---|---|---|---|
| 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 |

Example: `\dmatrix[lines,sep=0.75]{0,2,1,4,3,6,5}`

| | | | | | | |
|---|---|---|---|---|---|---|
| 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 |

## 2.2 \ddiagram

`\ddiagram` produces a twelve-tone clock diagram of arbitrary length, (as seen here).

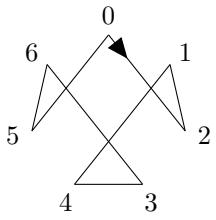
Options:

- up — specifies the number at the top (default is the first in the row).
- arrow shift — adjusts arrow position on the line (default is 2.5, range 0-10).
- name — adds a label at the center.
- no numbers — hides the surrounding numbers.
- no arrow — hides the arrow.
- no tikz — disables automatic TikZ environment, letting the user type it. It is recommended to pass ddiagram to the user's TikZ environment:

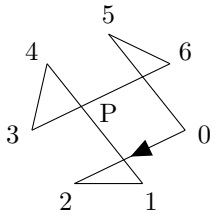
```
\begin{tikzpicture}[ddiagram]
\ddiagram[no tikz]{0,2,1,4,3,6,5}
\end{tikzpicture}
```

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

Example: `\ddiagram{0,2,1,4,3,6,5}`




Example: `\ddiagram[name=P, up=5, arrow shift=5]{0,2,1,4,3,6,5}`




## 2.3 \ddihedral

`\ddihedral` produces a "dihedral" representation of a series of arbitrary length (introduced in this article to illustrate the transformations' group structure).

Options:

- t — applies the transformation *transposition*.
- s — applies the transformation *inversion*.
- c — applies the transformation *cyclic shift*.
- v — applies the transformation *retrograde*.

These transformations are applied in this exact order: `t`, `s`, `c`, `v`.

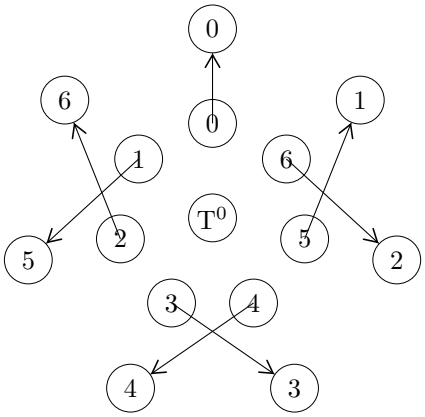
new t, new s, new c, new v  rename respective transformations on the label.

no tikz  disables automatic TikZ environment, letting the user type it. It is recommended to pass ddihedral to the user's TikZ environment:

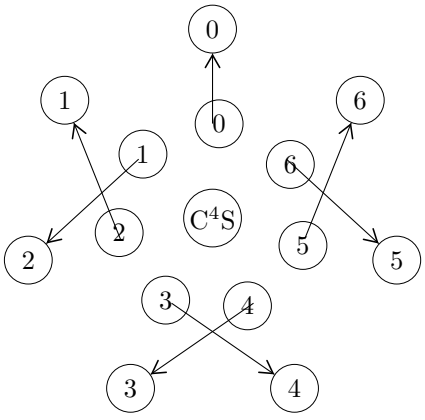
```
\begin{tikzpicture}[ddihedral]
\ddihedral[no tikz]{0,2,1,4,3,6,5}
\end{tikzpicture}
```

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

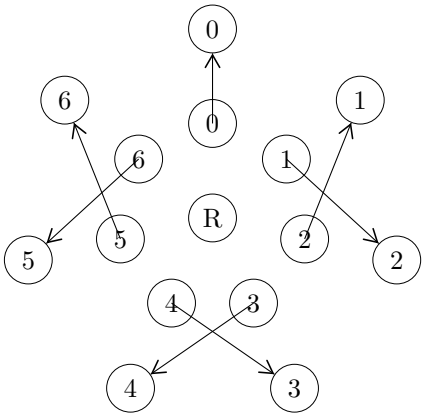
Example: `\ddihedral{0,2,1,4,3,6,5}`




Example: `\ddihedral[s=1, c=4]{0,2,1,4,3,6,5}`



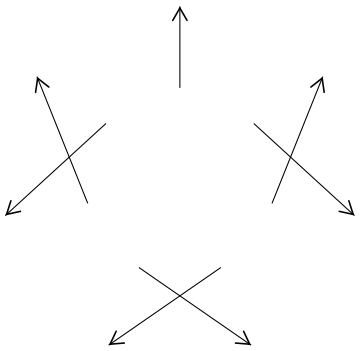

Example: `\ddihedral[new v=R, v=1]{0,2,1,4,3,6,5}`




`\darrows` draws the arrows from `\ddihedral` (which represent the row's orbit).

`no tikz` disables automatic TikZ environment, letting the user type it.

Example: `\darrows{0,2,1,4,3,6,5}`



## 2.4 \drow

`\drow` produces a twelve-tone row sequence as a mathematical permutation in its matrix form (using an *array* environment).

Options: `sep` adjusts column spacing.

Example: `\drow{0,2,1,4,3,6,5}`

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

Example: `\drow[sep=10pt]{0,2,1,4,3,6,5}`

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

# 3 Notes from the author

This package was created to support the articles written for the DivulgaMAT journal (in Spanish), under the column *Matemáticas y Música*:

- 100. (Septiembre 2019) Serialismo y matemáticas (I)
  - Introduces the `\dmatrix` and `\drow` commands.
- 101. (Octubre 2019) Serialismo y matemáticas (II)
  - Introduces the `\ddiagram`, `\darrows`, and `\ddihedral` commands. The `\ddihedral` command is here invented to illustrate the dihedral $\times$ dihedral group structure of the four historical transformations of a twelve-tone series.
- 102. (Noviembre 2019) Serialismo y matemáticas (III)
- 103. (Diciembre 2019) Re-escalando música

# 4 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.

\ProvidesPackage{ddphonism}
[2025/05/13 v0.3 Dodecaphonic diagrams: twelve-tone matrices, clock diagrams, etc.]

\RequirePackage{tikz}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Utilities

\newcounter{D@size}
\newcommand{\D@sizeMake}[1]{%
    \setcounter{D@size}{0}%
    \foreach \n in {#1} {\stepcounter{D@size}}%
}

\newcounter{D@head}
\newcommand{\D@headMake}[1]{%
    \setcounter{D@head}{-1}%
    \foreach \n in {#1}%
    {\ifnum\theD@head=-1\setcounter{D@head}{\n}\fi}%
}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Matrices
```

```
\usetikzlibrary {matrix}

\newif\ifD@matrixLines
\newif\ifD@matrixO
\newif\ifD@matrixI
\newif\ifD@matrixV
\newif\ifD@matrixH
\newif\ifD@matrixTikz
\pgfkeys{
    /dmatrix/.is family
    , /dmatrix
    , default/.style =
    { lines = false, outside lines = false, inside lines = false
    , sep = 1, vsep = 1, hsep = 1, no tikz = false }
    , lines/.is if=D@matrixLines
    , outside lines/.is if=D@matrixO
    , inside lines/.is if=D@matrixI
    , vlines/.is if=D@matrixV
    , hlines/.is if=D@matrixH
    , sep/.estore in=\D@matrixSep
    , vsep/.estore in=\D@matrixVsep
    , hsep/.estore in=\D@matrixHsep
    , no tikz/.is if=D@matrixTikz
}

\newcommand{\D@LOH}{% outside horizontal lines
    \foreach \y in {0, {-0.5*\theD@size*\D@matrixSepVsep}}
        \draw (0,\y) -- ({\theD@size*\D@matrixSepHsep},\y);%
}

\newcommand{\D@LOV}{% outside vertical lines
    \foreach \x in {0, {\theD@size*\D@matrixSepHsep}}
        \draw (\x, 0) -- (\x, {-0.5*\D@matrixSepVsep*\theD@size});%
}

\newcommand{\D@LIH}{% inside horizontal lines
    \pgfmathparse{\theD@size - 1}\foreach \x in {1,...,\ pgfmathresult}
        \draw (0,{-0.5*\x*\D@matrixSepVsep}) --%
        ({\theD@size*\D@matrixSepHsep},{-0.5*\x*\D@matrixSepVsep});%
}

\newcommand{\D@LIV}{% inside vertical lines
    \pgfmathparse{\theD@size - 1}\foreach \x in {1,...,\ pgfmathresult}
        \draw ({\x*\D@matrixSepHsep},0) --%
        ({\x*\D@matrixSepHsep},{-\theD@size*0.5*\D@matrixSepVsep});%
}

\newcommand{\dmatrix}[2][]{%
    \pgfkeys{/dmatrix, default, #1}%
    \D@sizeMake{#2}\D@headMake{#2}%
    \pgfmathsetmacro{\D@matrixSepVsep}{\D@matrixSep*\D@matrixVsep}%
    \pgfmathsetmacro{\D@matrixSepHsep}{\D@matrixSep*\D@matrixHsep}%
    \ifD@matrixTikz\else\begin{ tikzpicture }\ fi%
    \foreach [count=\nj] \j in {#2}
        \foreach [count=\ni] \i in {#2} {%
            \pgfmathsetmacro{\D@matrixI}%
                {int(mod(\i - \j + \theD@head + \theD@size, \theD@size))}%
```

```
            \draw node at ({(\ni−0.5)∗\D@matrixSepHsep},%
                {−0.5∗(\nj−0.5)∗\D@matrixSepVsep})%
            {\D@matrixl};%
        }%
    \ifD@matrixLines\D@LOH\D@LOV\D@LIH\D@LIV\fi%
    \ifD@matrixV\D@LOV\D@LIV\fi%
    \ifD@matrixH\D@LOH\D@LIH\fi%
    \ifD@matrixO\D@LOH\D@LOV\fi%
    \ifD@matrixI\D@LIH\D@LIV\fi%
    \ifD@matrixTikz\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]
\ tikzset {D@invclip/. style ={clip, insert  path={{[reset cm]
      (−16383.99999pt,−16383.99999pt) rectangle (16383.99999pt,16383.99999pt)}}}}

\newif\ifD@diagramTikz
\newif\ifD@diagramNoNum
\newif\ifD@diagramNoArr
\pgfkeys{
    /ddiagram/.is  family
    ,  /ddiagram
    ,  default /. style  =
    {  up =\empty, name =\empty, no tikz = false
    ,  no numbers = false,  no arrow = false ,  arrow  shift  = 2.5 }
    ,  no  tikz /. is   if =D@diagramTikz
    ,  no numbers/.is  if =D@diagramNoNum
    ,  no arrow/. is   if =D@diagramNoArr
    ,  name/.estore  in=\D@diagramName
    ,  up/. estore  in=\D@diagramUp
    ,  arrow  shift /. estore  in=\D@diagramArrS
}

\newcounter{D@prev}
\newcommand{\ddiagram}[2][]{%
    \pgfkeys{/ddiagram,  default ,  #1}%
    \D@sizeMake{#2}\D@headMake{#2}%
    \pgfmathsetmacro{\D@up}%
        { int (\ ifx \D@diagramUp\empty\theD@head\else\D@diagramUp\fi)}%
    %
    \ifD@diagramTikz\else\begin{ tikzpicture }[ddiagram]\ fi %
    \begin{scope}[rotate=360∗\D@up/\theD@size]%
    \ ifx \D@diagramName\empty\else%
    \node at  (0,0)  [ circle ]  {\D@diagramName};%
    \begin{pgfinterruptboundingbox}%
        \path[D@invclip]  (0,0)   ellipse   %
            ({0.02∗width("\D@diagramName")} and {0.02∗height("\D@diagramName")});%
    \end{pgfinterruptboundingbox}%
    \ fi
    %
    \pgfmathparse{\theD@size − 1}\foreach \x in  {0,...,\ pgfmathresult}  {% numbers
        \ifD@diagramNoNum\else\node at ({90−360∗\x/\theD@size}:2) {\x};\fi%
```

```
            \coordinate (\x) at ({90-360*\x/\theD@size}:1.6);%
        };
        \setcounter{D@prev}{-1}%
        \foreach \x in {#2}{% lines
            \ifnum\theD@prev=\theD@head% second
                \draw [decoration={markings,mark=at position 0.099*\D@diagramArrS with
                {\arrow[scale=1.25,>=triangle 45]{>}}},postaction={decorate}] (\theD@prev) -- (\x);%
            \else\ifnum\theD@prev>-1\draw (\theD@prev) -- (\x);\fi\fi% third onward
            \setcounter{D@prev}{\x}%
        };%
        \draw (\theD@prev) -- (\theD@head);% last
        \end{scope}%
        \ifD@diagramTikz\else\end{tikzpicture}\fi%
    }


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

    \tikzstyle{D@dihedralArrow}=
        [decoration={markings,mark=at position 1 with
        {\arrow[scale=1.5,>=angle 60]{>}}},postaction={decorate}]
    \tikzstyle{ddihedral}=[inner sep=0,minimum height=18pt]

    \newif\ifD@dihedralTikz
    \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}
        , t/.estore in = \D@dihedralT
        , c/.estore in = \D@dihedralC
        , s/.estore in = \D@dihedralS
        , v/.estore in = \D@dihedralV
        , new t/.estore in = \D@dihedralNewT
        , new c/.estore in = \D@dihedralNewC
        , new s/.estore in = \D@dihedralNewS
        , new v/.estore in = \D@dihedralNewV
        , no tikz/.is if=D@dihedralTikz
    }

    \newif\ifdarrowsTikz
    \pgfkeys{
        /darrows/.is family, /darrows,
        default/.style = {no tikz=false},
        no tikz/.is if=darrowsTikz,
    }

    \newcommand{\darrows}[2][]{%
        \pgfkeys{/darrows, default, #1}%
        \D@sizeMake{#2}%
        \ifdarrowsTikz\else\begin{tikzpicture}\fi%
        \pgfmathparse{\theD@size - 1}\foreach \x in {0,...,\pgfmathresult}%
            \draw ({90-360*\x/\theD@size}:2.5) node[circle] (\x) {};%
        \foreach \x [count=\y] in {#2}%
            \draw[style=D@dihedralArrow] ({90-360*(\y-1)/\theD@size}:1.25) -- (\x);%
        \ifdarrowsTikz\else\end{tikzpicture}\fi%
```

```
    }
205
    \newcommand\ddihedral[2][]{%
207     \pgfkeys{/ddihedral, default, #1}%
        \D@sizeMake{#2}%
209     %
        \ifD@dihedralTikz\else\begin{tikzpicture}[ddihedral]\fi%
211         \def\D@dihedralName{%
            \ifodd\D@dihedralV{\D@dihedralNewV}\else%
213         \ifnum\D@dihedralC=0%
            \ifodd\D@dihedralS\else%
215         \ifnum\D@dihedralT=0{\D@dihedralNewT$^0$}%
            \fi\fi\fi\fi%
217         \ifnum\D@dihedralC=0\else\D@dihedralNewC$^{\D@dihedralC}$\fi%
            \ifodd\D@dihedralS{\D@dihedralNewS}\fi%
219         \ifnum\D@dihedralT=0\else\D@dihedralNewT$^{\D@dihedralT}$\fi%
            }
221     \node at (0,0) [very thin,draw,circle,inner sep=1pt] {\D@dihedralName};%
        \begin{pgfinterruptboundingbox}%
223         \path[D@invclip] (0,0) circle %
                ({0.02*width("\D@dihedralName")} and {0.02*height("\D@dihedralName")});%
225     \end{pgfinterruptboundingbox}%
        \pgfmathparse{\theD@size - 1}\foreach \x in {0,...,\pgfmathresult} {%
227         \draw ({90+(\D@dihedralT + (2*\D@dihedralS-1)*\x)*360/\theD@size}:2.5)%
            node[very thin, circle,draw] {\x};%
229         \draw ({90-(\D@dihedralC+(2*\D@dihedralV-1)*\x*360/\theD@size)}:1.25)%
            node[very thin, circle,draw] {\x};%
231     }%
        \darrows[no tikz]{#2}%
233     \ifD@dihedralTikz\else\end{tikzpicture}\fi%
    }
235

237 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Rows
239
    \pgfkeys{
241     /drow/.is family, /drow,
        default/.style = {sep=\arraycolsep},
243     sep/.estore in = \D@rowSep,
    }
245
    \long\def\D@concat#1#2{\expandafter\def\expandafter#1\expandafter{#1#2}}
247 \newlength{\D@ogsep}
    \newcommand{\drow}[2][]{%
249     \pgfkeys{/drow, default, #1}%
        \D@sizeMake{#2}%
251     \setlength{\D@ogsep}{\arraycolsep}\setlength{\arraycolsep}{\D@rowSep}%
        %
253     \ifnum\theD@size=0{\ensuremath{\left(\right)}}%
        \else\ifnum\theD@size=1
255     \ensuremath{\left(\begin{array}{*{\theD@size}c}0\\0\\\end{array}\right)}%
        \else%
257     \global\def\D@firstrow{}\global\def\D@secondrow{}%
        \foreach \x [count=\i from 0] in {#2} {%
259         \ifnum\i>0
                \xdef\D@firstrow{\D@firstrow & \i}
```

```
261                 \xdef\D@secondrow{\D@secondrow & \x}%
            \else
263                 \xdef\D@firstrow{\i}
                    \xdef\D@secondrow{\x}
265             \fi%
        }%
267     \ensuremath{\left(\begin{array}{*{\theD@size}c}%
                    \D@firstrow \\ \D@secondrow \\ \end{array}\right)}%
269     \fi%
        \setlength{\arraycolsep}{\D@ogsep}%
271 }

273 \endinput

275
    %% End of file `ddphonism.sty'.
```