MusicTEX: using TEX to write polyphonic or instrumental music

Daniel TAUPIN†

† *Université de Paris-Sud*
Laboratoire de Physique des Solides
bâtiment 510
F-91405 ORSAY Cedex, France

Abstract

MusicTEX is a set of TEX or LaTEX macros which are fit to typeset polyphonic, instrumental or orchestral music. It is able to handle an important number of instruments or voices (up to nine) and staffs (up to four for each instrument). Many usual ornaments have been provided, including several note sizes which can handle grace notes or extra music like cadenzas.

Except the risk of typing errors due to a sophisticated set of macros, the major difficulty still resides in glue and line breaking in the case of irregular music and slurs.

Résumé

MusicTEX est un jeu de macro-instructions TEX ou LaTEX adaptées à l'écriture de partitions polyphoniques, instrumentales ou orchestrales. MusicTEX est capable de gérer un nombre important d'instruments ou de voix (jusqu'à neuf) et de portées (jusqu'à quatre par instrument). En outre, beaucoup d'ornements sont maintenant disponibles, notamment les notes de petite taille our indiquer les ornements explicites ou les cadences.

À part les risques d'erreurs liées à la difficulté d'usage de certaines procédures, la principale difficulté est celle des coupures de lignes dans le cas où d'une musique irrégulière et la rálisation de liaisons ou de phrasés esthétiques.

Key words: music typesetting, orchestral music, polyphonic music.

1 What is MusicTEX ?

MusicTEX is a set of TEX macros to typeset polyphonic, orchestral or polyphonic music. Therefore, it is mainly supposed to be used to type wide scores – just because true musicians seldom like to have to frequently turn pages – and this is not really compatible with LaTEX's standard page formats, even the `A4.sty` whose `\textheight` and `\textwidth` are too small for musician needs.

However, a LaTEX style has been also provided (and it is used for the typing of the present paper) but this `musictex` style is fit for musicographic books rather than for normal scores to be actually played.

It is to be emphasized that MusicTEX is not intended to be a compiler which would translate into TEX some standard musical notations, nor to decide by itself about aesthetic problems in music typing. MusicTEX only typesets staves, notes, chords, beams, slurs and ornaments as requested by the composer. Since it makes very few typesetting decisions, MusicTEX appears to be a versatile and rather powerful tool. However, due to the important amount of informations to be provided to the typesetting process, coding MusicTEX might appear to be awfully complicated, just as the real keyboard or orchestral music. It should be interfaced therefore by some pre-compiler in the case of the composer/typesetter wanting aesthetic decisions to be automatically made by somebody (or something) else.

2 MusicTEX pricipal features

2.1 Music typesetting is two-dimensional

Most of the people who just learnt a bit of music at college probably think that music is a linear sequence of symbols, just as literary texts to be TEX-ed. In fact, with the exception of strictly monodic instruments like most orchestral wind instruments and solo voices, one should be aware that reading music actually is a matricial operation : the non-soloist musician successively reads *columns* of simultaneous notes which he actually plays if he is a pianist, clavichordist or organist, which he actually reads and watches if he conducts an orchertra, and which he is supposed to check and partially play when he is a soloist who wants to play in time with the accompanying instrument or choir.

In fact, our personal experience of playing piano and organ as well as sometimes helping as an alternate Kapellmeister leads us to think that actual music reading and composing is a slightly more complicated intellectual process : music reading, music composing and music thinking seems to be a three layer process. The musician usually reads or thinks several consecutive notes (typically a long *beat* or a group of logically connected notes), then he

note sequence one	*note seq. four*	*note seq. seven*	*note seq. ten*
note sequence two	*note seq. five*	*note seq. eight*	*note seq. eleven*
note seqence three	*note seq. six*	*note seq. nine*	*note seq. twelve*

Table 1: The order in which a musician reads music

goes down to the next instrument or voice and finally assembles the whole to build a part of music lasting roughly a few seconds. Then he handles the next *beat* or *bar* of his score.

Thus, it appears that the humanly logical way of coding music consists in horizontaly accumulating a set of *vertical combs* with *horizontal teeth* as described in Table 1.

This is the reason why, in **MusicTEX** the fundamental *macro* is of the form

```
\notes ... & ... & ... \enotes
```

where the character `&` is used to separate the notes to be typeset on respective staffs of the various instruments, starting from the bottom.

In the case of an instrument whose score has to be written with several staffs, these staffs are separated by the character `|`. Thus, a score written for a keyboard instrument and a monodic instrument (for example piano and violin) will be coded as follows :

```
\notes ... | ... & ... \enotes
```

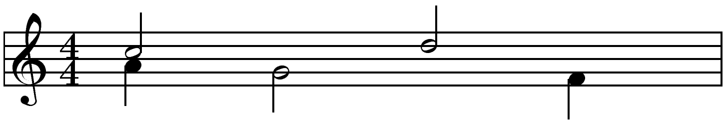
for each column of simultaneous *groups of notes*. It is worth emphasizing that we actually said *"groups of notes"*: this means that in each section of the previous macro, the music typesetter is welcome to insert, not only chord notes to be played at once, but small sequences of consecutive notes which build something he understands as a musical phrase. This is why note typing macros are of two kinds in MusicTEX, namely the note macros which are not followed by spacing afterwards, and those which induce horizontal spacing afterwards.

2.2 The spacing of the notes

It seems that many books have dealt with this problem. Although it can lead to interesting algorithms, we think it is in practice a rather minor one.

In fact, each column of notes has not necessarily the same spacing and, in principle, this *spacing* should depend on the shortest duration of the simultaneous notes. But this cannot be established as a rule, for at least two reasons :

1. spacing does not depend only of the local notes, but also on the context, at least in the same bar.
2. in the case of polyphonic music, exceptions can easily be found. Here is an example :

   where it can be clearly seen that the half notes at beats 2 and 3 must be spaced as if they were quarter notes since they overlap, which is obvious only because of the presence of the indication of the *meter* 4/4.

Therefore, we prefered to provide the composer/typesetter with a set of macros, the spacing of which increases by a factor of $\sqrt{2}$ (incidentally, this can be adjusted) :

`\notes ... & ... & ... \enotes`	`%` 1 spatial unit
`\Notes ... & ... & ... \enotes`	`%` 1.4 spacial unit
`\NOtes ... & ... & ... \enotes`	`%` 2 spatial units
`\NOTes ... & ... & ... \enotes`	`%` 2.8 spatial units
`\NOTEs ... & ... & ... \enotes`	`%` 4 spatial units
`\NOTES ... & ... & ... \enotes`	`%` 5.6 spatial units

The size of the spatial unit (`\elemskip`) can be freely adjusted. In addition, MusicTEX provides a means of adjusting the note spacing according to an average number of elementary spaces within a line (macro `\autolines`).

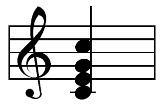
2.3 Music tokens, rather than a readymade generator

The tokens provided by MusicTEX are :

- the note symbols *without stem* ;
- the note symbols *with stems, and hooks for eighth notes and beyond* ;
- the indications of beam beginnings and beam ends ;
- the indications of beginnings and ends of ties and slurs ;
- the indications of accidentals ;
- the ornaments : arpeggios, trills, mordents, pincés, turns, staccatos and pizzicatos, fermatas ;
- the bars, the meter and signature changes, etc.

Thus, `\wh g` produces an *A (445 Hz)* whose duration is a *whole note*. In the same way, `\qu c` produces a *C (250 Hz approx.)* whose value is a *quarter note with stem up*, `\cl J` produces a *C (125 Hz approx.)* whose duration is an *eighth note with stem down*, etc.

To generate quarter, eighth, sixteenth, etc. chords, the macro `\zq` can be used : it produces a quarter note head whose position is memorized and recalled when another stemmed note (possibly with a hook) is coded ; then the stem is adjusted to link all simultaneous notes. Thus, the perfect C-major chord, i.e.



is coded `\zq c\zq e\zq g\qu j` or, in a more concise way, `\zq{ceg}\qu j` (stem up) : in fact, single notes are treated... like one-note chords.

2.4 Beams

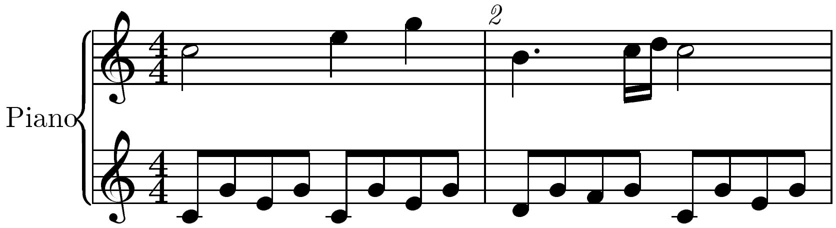
Beams are generated using macros which define their beginning (at the current horizontal position), together with their altitude, their sense (upper of lower), their multiplicity, their slope and their reference number. This latter feature – the reference number – appears to be necessary, since one may want to write beams whose horizontal extents overlap : therefore, it is necessary to specify which beam the notes hang on and which beam is terminated at a given position.

2.5 Setting anything on the score

A general macro (`\zcharnote`) provides a means of putting any sequence of symbols (in fact, some `\hbox{...}`) at any pitch of any staff of any instrument. Thus, any symbol defined in a font (letters, math symbols, etc.) can be used to typeset music.

2.6 A simple example

Before entering other details, we give below an example of the two first bars of the sonata in C-major K545 by MOZART :



The *coding* is set as follows :

```
\begin{music}
\parindent 1cm
\def\nbinstruments{1}\relax % a single instrument
\def\instrumenti{Piano}%    % whose name is Piano
\nbporteesi=2\relax         % with two staffs
\generalmeter{\meterfrac{4}{4}}\relax % 4/4 meter choosen
\debutextrait               % starting real score
\normal                     % normal 12 pt note spacing
\temps\Notes\ibu0f0\qh0{cge}\tbu0\qh0g|\hl j\enotes
\temps\Notes\ibu0f0\qh0{cge}\tbu0\qh0g|\ql l\sk\ql n\enotes
\barre                      % bar
\Notes\ibu0f0\qh0{dgf}|\qlp i\enotes
\notes\tbu0\qh0g|\ibbl1j3\qb1j\tbl1\qb1k\enotes
\temps\Notes\ibu0f0\qh0{cge}\tbu0\qh0g|\hl j\enotes
\finextrait                 % terminate excerpt
```

```
\end{music}
```

- `\ibu0f0` begins an upper beam, aligned on the f, reference number 0, slope 0
- `\tbu0` terminates this beam before writing the second g by means of `\qh0g`
- `\qh..` indicates a note hanging on a beam.
- `\sk` sets a space between the two quarters at the right hand, so that the second is aligned with the third eighth of the left hand.
- `\qlp` is a quarter with a point.
- `\ibbl1j3` begins a double beam, aligned on the C (`j` at this pitch) of slope 0.15.

2.7 Signatures

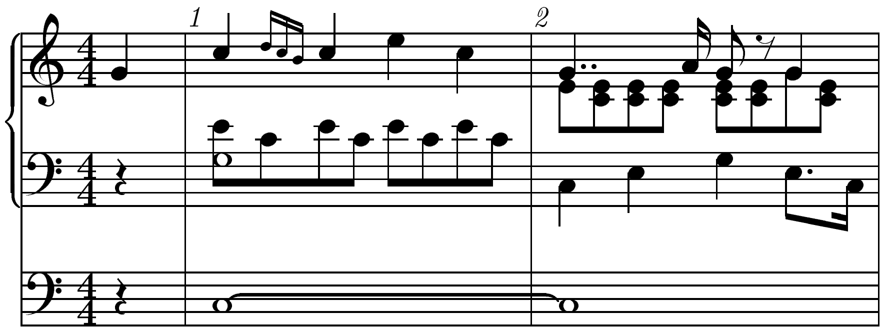
Signatures are stated either for all instruments, such as : `\generalsignature=-2` which sets two flats on each staff, but this can be partly overrided by `\signatureii=1` which puts one sharp on the staffs of *instrument number* 2 (ii). Of course, the current signature may change at any time as well as meters and clefs.

2.8 Transposition

An important question is : *"can MusicTEX transpose a score ?"*. The answer is now 99.5 % *yes*. If fact, there is an internal register named `\transpose` the default value of which is zero, but it may be set to any positive of negative reasonable value. Then, it offsets all symbols pitched with letter symbols by that number of pitch steps. However, it will neither change the signature nor the local accidentals, and if – for example – you transpose by 1 pitch a piece written in C, MusicTEX will not know whether you want it in $D\flat$, in D or in $D\sharp$. This might become tricky if accidentals occur within the piece, which might have to be converted into flats, naturals, sharps or double sharps, depending on the new choosen signature. To circumvent this trouble, *relative* accidentals have been implemented, the actual output of which depends of the pitch of this accidental and of the current signature.

2.9 Grace notes and cadenzas

In addition to its facility of generating either sixteen point or twenty point staffs with note heads of corresponding size, MusicTeX also allows the user to type smaller notes, in order to represent either *grace notes*, *cadenzas* or a proposed realization of a *figured bass*. This may give something like :



2.10 Selecting special instrument scores

Another question is : *"can I write an orchestral score and extract the separate scores for individual instruments ?"* The answer is 95 % *yes* : in fact, you can define your own macros `\mynotes...\enotes`, `\myNotes...\enotes` with as many arguments as there are in the orchestral score (hope this is less or equal to 9, but TeXperts know how to work around) and change its definition depending on the selected instrument (or insert a test on the value of some selection register). But the limitation is that the numbering of instruments may change, so that `\signatureiii` may have to become `\signaturei` if instrument *iii* is alone. But, in turn, this is not a serious problem for average TeX wizard apprentices.

3 How to get it

The whole *distribution* fits into a single 1.2Mbyte or 1.44Mbyte diskette. It can also be obtained through an *anonymous ftp* at `rsovax.ups.circe.fr` (130.84.128.100), after selecting the subdirectory `[.musictex]`. All sources (including fonts) are provided, either separately or "zipped" or as VMS 'savesets".

4 Implementation, restrictions and enhancement possibilities

The macroinstruction file MusicTEX contains approximately 2500 lines of code, that is 80 000 bytes approximately. This requires your score to be compiled by the most extended versions of TEX (65 000 words of working memory). In desperate situations, we recommend using the "BigTEX" processors which, unfortunately, perform a great deal of disk input/outputs (on PCs) which make them awfully slow.

In particular, the number of registers it uses can hinder its compatibility with some LaTEX styles or with LaTEX itself in case of restricted memory availability.

4.1 Recent easy enhancements

Many enhancements have been asked for, and this is a proof thet MusicTEX is considered as useful by many people. Some of these enhancements which seemed hard were in fact rather easy to implement, for example small notes to represent grace notes and cadenzas. But others may induce heavy problems, for example the need of having *nice* slurs and ties.

4.2 The tie/slur problem

While typesetting notes and even beams is a rather simple problem because is is a *local typesetting*, ties and slurs are much more difficult to handle.

Or course there is little problem in case of a typesetter wanting a slur or a tie binding two consecutive notes, not separated by a bar. In practice this *very restricted* use of slurs or ties can easily be solved by putting some symbols extracted from the `slur16` or `slurn16/slurn20` fonts somewhere on the staffs using the general use `\zcharnote` macro.

But serious music typesetters or composers know that many ties are supposed to link notes which are on both sides of a bar, which is a likely place to insert line breakings, so that the *tie* coding must have various versions and sizes to resist that possible line breaking. What has been said about ties is still more serious in the case of *phrasing slurs* which may extend over several bars, lines and sometimes pages. In this case, their shape is not only a question of producing a long curved symbol of nice looking shape, it also has to cope with *glue*. An then the worst is that music way of typing does not accept *ragged lines* but equal length lines, even for the last line of a music piece. Thus, long distance slurs and ties need to be cut into separate parts (beginning, continuing(s), endings) which TEX can only link using *horizontal line overlaps* or `\leaders` to insure slur continuity over this unavoidable glue.

Therefore and up to now, ties and slurs have been implemented in a way which may look rather ugly, but we think it is the only way of implementing *in one pass* ties and slurs which run *across glue*. The principle is to have tie/slur symbols with a rather long part of horizontal stuff. Then, at each time a glue occurs and at each time a group of notes is coded while a slur or tie is pending, an `\hrule` is issued which overlaps the preceeding tie/slur symbol so that the final output seems to contain a continuous line. Unfortunately, this is possible only in the glue expansion direction, namely in the horizontal direction.

It could be requested – and this is thought of – to have variable size initial and final curved slur symbols which the user would choose according to his intention to have short or long range slur symbols. This has not been *yet* implemented for several reasons :

- The author's natural lazyness (!)
- More seriously : this would require using some more registers (at least a dozen) to record the initial sizes (horizontal and vertical) of this symbol *for each slur/tie* in order to make adequate links over glue and to close it with the symmetrical symbol. Unfortunately, TEX registers are not numerous (256 of each kind) and we are afraid such a feature would make TEX stupidly crash even when typesetting reasonable scores.
- We do not think it wise to introduce in MusicTEX itself a great number of macros which would be pooly used by most users : the reason is that TEX memory is hardly limited and that unused macros may occupy some TEX storage which could make things crash because of `TeX capacity exceeded`...

5 Acknowledgements

The idea of implementing the present package is due to the previous work (MuTEX) of Andrea Steinbach and Angelika Schofer[1]. This work provided the basis of the Metafont codes and some line breaking procedures, which both are still used here... with 99% corrections and updates.

6 Some examples

Many examples can be typeset from the MusicTEX distribution, but they are not included here because of their need to be compiled with TEX rather than LaTEX and because they need output paper of the extended 291×297 mm^2 european size. However we chose to produce a small type size version of HAYDN's *aria* from the Creation and a piece of a personal composition heavily using beams.

References

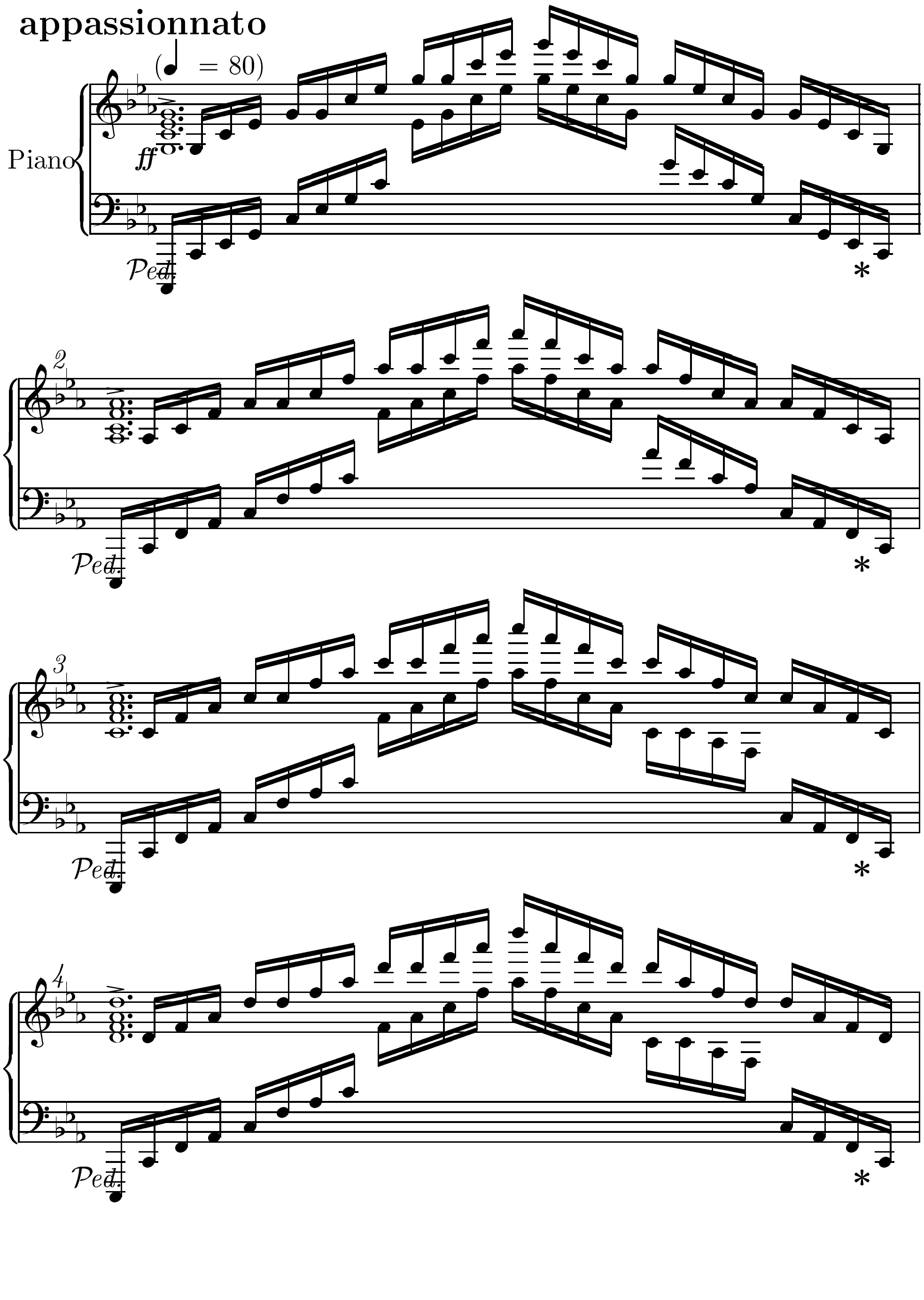
[1] Andrea STEINBACH and Angelika SCHOFER, *Theses* (1987, 1988), Rheinische Friedrich-Wilhelms Universität, Bonn, Germany.

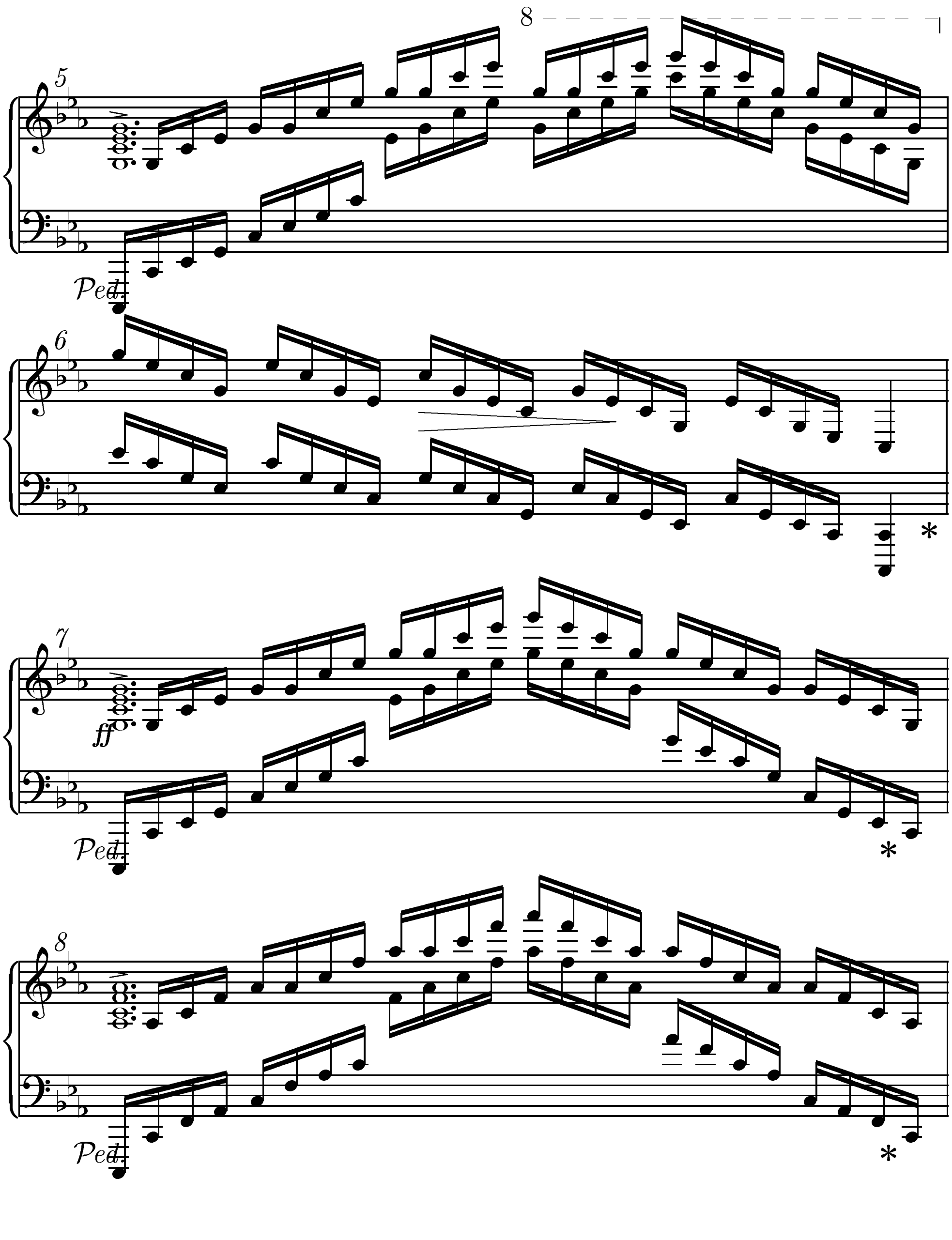
Aria No. 24
(The Creation)

Joseph Haydn

Transcription for organ and tenor, D. Taupin (1990)


9
10
11
12
Mit Würd und Ho- heit
II
I
13
14
an- ge- tan, mit Schön- heit, Stärk und


appassionnato
(♩ = 80)
Piano
ff
Ped.
*
2
Ped.
*
3
Ped.
*
4
Ped.
*


8
5
Ped.
6
*
7
ff
Ped.
*
8
Ped.
*