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
taupin@rsovax.ups.circe.fr

Abstract

MusicTEX is a set of TEX or LATEX macros — initially posted three years ago and now used by dozens of music typesetters — 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 of the usual ornaments have been provided, including several note sizes which can handle grace notes or extra music like cadenzas.

A recent enhancement consisted of providing a facility for several staff sizes in the same score, thus enabling full size staffs to smaller "reminding" staffs.

The LATEX version is not really fit for printing full scores but it has be used to produce musicographic texts including many (small but numerous) music excerpts.

Except for 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.

What is MusicTEX?

Several packages exist which provide the personal computer addict a facility for typesetting music. For instance we saw examples from *Personal Composer* and *Musictime*, and we experimented first with the MUTEX package (see Steinbach and Schofer, 1987, 1988), the latter being based on TEX and METAFONT. However, all these packages have limitations: either the output quality (*Personal Composer*) or the complexity of the score (*Musictime*), or the number of staffs (MUTEX).

Thus, a few years ago, we could not resist the temptation of building a new package — in fact a set of TEX macros and fonts — which would be able to typeset complex polyphonic, orchestral or instrumental music. In fact our primary intention was to extend MUTEX to several staffs, but was quickly apparent that rewriting the whole of the macros was a better solution and we only used MUTEX's METAFONT code as a starting point.

Although not perfect, MusicTEX appears to be a powerful tool which can handle up to nine distinct instruments, each having from zero (for lyrics) to four staffs. Of course it can handle chords or polyphonic note settings in the same staff and we have used it to typeset realistic music for choirs and instruments, including organ.

It must 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 engraver. Since it makes very few typesetting decisions, MusicTEX appears to be a versatile and rather powerful tool, but in turn it should be interfaced by some precompiler for the engraver who wants aesthetic decisions to be automatically made by somebody (or something) else.

One can also mention a secondary use of MusicTEX as a *target* language for music coding, namely the MIDI2TeX package by Hans Kuykens, which translates MIDI data files into MusicTEX source code (Kuykens, 1991). Notwithstanding capacity problems, a LATEX style has also been provided (it was used to typeset the present paper) but this `musictex` style is fit for musicographic books rather than for normal scores to be actually played.

MusicTEX principal features

Music typesetting is two-dimensional. Most of the people who just learned 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 is actually a matricial operation: a musician playing a chordal instrument — guitar, piano, organ — or looking at more than one staff — a choir singer, a conductor — successively reads *columns* of simultaneous notes which he or she plays or at least watches in order to be in time with the others.

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

Thus, it appears that the most logical way of coding music consists of horizontally accumulating a set of *vertical combs* with *horizontal teeth* as described below:

sequence one	*seq. four*	*seq. seven*
sequence two	*seq. five*	*seq. eight*
sequence three	*seq. six*	*seq. nine*

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

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

where the character & is used to separate the notes (or the groups of notes) to be typeset on the 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 or single staff instrument (for example piano and violin) will be coded as follows:

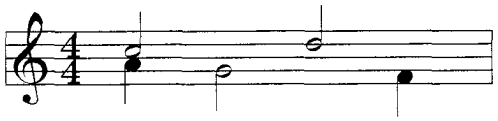
```
\notes ... | ... & ... \enotes
```

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.

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 does not necessarily have 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 on 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 preferred providing the engraver with a set of macros having specific spacings (`\noteskip`) whose ratio to a general basic spatial unit `\elemskip` increases by a factor of $\sqrt{2}$ (incidentally, this can be adjusted):

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

The size of both the basic spatial unit (`\elemskip`) and the note-specific spacing (`\noteskip`) can be freely adjusted since they are not `\global`. In addition, MusicTEX provides a means of adjusting the basic spacing `\elemskip` according to an average number of elementary spaces within a line (macro `\autolines`).

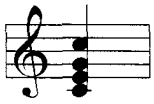
Music tokens, rather than a readymade generator. The tokens provided by MusicTEX are:

- the note symbols *without stems*;
- 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.

As an example, a half note of pitch *A* (the *A* at the top of the bass clef) with stem up is coded as `\hu a` and all pitches above that *A* are represented with lowercase letters up to z; uppercase letters represent grave notes, i.e.; those usually written under the bass clef. In the same way `\wh h` produces an *A* (the one in the middle of the G clef staff, i.e., 445 Hz approx.) whose duration is a *whole note*, `\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.

It is worth pointing out that pitch coding in MusicTEX is related to the actual note pitch, not to the note head position under a given clef. Thus, if the typesetter wants to change the active clef of a part of the score, he doesn't have to change the pitch codings, perhaps only the sense of the stems and of the beams.

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.,

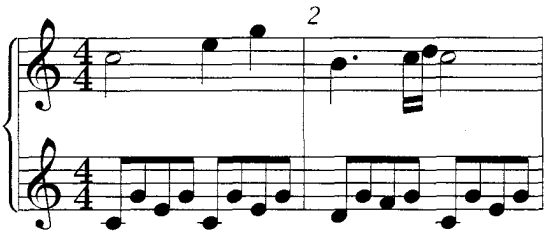


is coded `\zq c\zq e\zq g\qu j` or, in a more concise way, `\zq{ceg}\qu j` (stem up): in fact, single notes are treated... like one-note chords.

Beams. *Beams* are generated using macros which define their beginning (at the current horizontal position), together with their altitude, their sense (upper or 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.

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.

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}
\def\nbinstruments{1}\relax % single instrument
\nbporteesi=2\relax         % with two staffs
\generalmeter{\meterfrac44}\relax % 4/4 meter
\debutextrait               % start real score
\etroit                     % 10pt note spacing
\temps
\Notes\ibu0f0\qh0{cge}\tbu0\qh0g|\hl j\enotes
\temps
\Notes\ibu0f0\qh0{cge}\tbu0\qh0g\relax
                          |\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                 % end excerpt
\end{music}
```

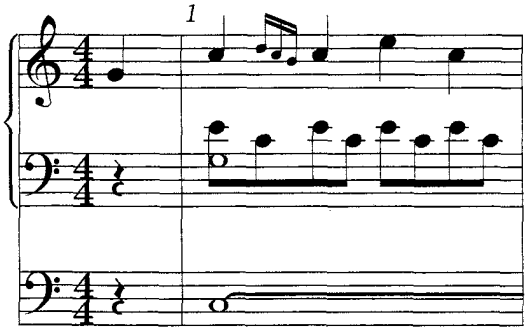
- `\ibu0f0` begins an upper beam, aligned on the *f*, reference number 0, slope 0.
- `\tbu0` terminates this beam before writing the second *g* by means of `\qh0g`.
- `\qh..` indicates a note hanging on a beam.
- `\sk` sets a space between the two quarters at the right hand, so that the second is aligned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

Signatures. Signatures are usually stated for all instruments, such as: `\generalsignature=-2` which

sets two flats on each staff; however this global signature can be partly overridden by instrument specific statements such as: `\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.

Transposition. Provided some precaution is taken concerning the accidentals, MusicTEX can transpose or partly transpose a score. In fact, there is an internal register named `\transpose`, the default value of which is zero, but which may be set to any reasonable positive or negative 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 chosen signature. To circumvent this trouble, *relative* accidentals have been implemented, the actual output of which depend on the pitch of this accidental and of the current signature.

Grace notes and cadenzas. In addition to its facility for 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:



Selecting special instrument scores. A frequent 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 (one hopes this is less than or equal to 9, but TEXperts know how to work around it) and change their 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.

How to get it

The whole *distribution* fits on a single 1.2Mbyte or 1.44Mbyte diskette. It is also available on anonymous `ftp` server `rsovax.ups.circe.fr` (130.84.128.100), after selecting the subdirectory `[anonymous.musictex]`. Several other `ftp` sites also provide it, especially `ftp.tex.ac.uk` and `ftp.gmd.de`. All sources are provided, including fonts. It can also be automatically e-mailed (uuencoded) by means of the message `SENDME MUSICTEX` sent to `FILESERV@SHSU.BITNET`.

Implementation

The macro file MusicTEX contains approximately 2500 lines of code, approximately 80 000 bytes. This requires your score to be compiled by the most extended versions of TEX (65 000 words of working memory), or with "BigTEX" processors which are unfortunately slow on 286 PCs, due to a great deal of disk input/output.

In particular, notwithstanding the fact that a great many dimension registers have been moved to `\fontdimen` registers — an ugly but efficient way of doing it — the number of registers it uses can hinder its compatibility with some LATEX styles or with LATEX itself in case of restricted memory availability.

Recent easy enhancements. Many enhancements have been asked for, and this is proof that MusicTEX is considered 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 or narrow staffs to represent informational score — not to be played — like the violin part above the actual piano staffs to be played by the reader. But others may induce heavy problems, for example, the need of having *nice* slurs and ties.

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

Or course there is little problem in the 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 engravers 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 breaks, 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*. Unfortunately, the way of typing music 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, 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, each time glue occurs and each time a group of notes is coded while a slur or tie is pending, an \hrule is issued which overlaps or links to 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.

A recent enhancement consisted of providing two kinds of slur macros (\ilegu*np* and \Ilegu*np*, same for lower slurs) to have variable size initial and final curved slur symbols which the user can choose according to his intention to have short or long range slur symbols.

Extensive slur size variations have not been implemented for several reasons:

- The lack of dimension registers (256 available are nearly exhausted in LATEX+MusicTEX) 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.
- We do not think it wise to introduce in MusicTEX itself a great number of macros which would be little used by most users and would overload the restricted TEX memory, resulting in too many TeX capacity exceeded crashes.

The drawback of complexity. Due to the large amount of information to be provided for the typesetting process, coding MusicTEX sometimes appears to be awfully complicated to beginners, just as does the real keyboard or orchestral music. This is a necessary inconvenience to achieve its power and we can only encourage people just wanting to typeset a single voice tune to ask their local TEX guru for a set of simplifying macros...

Some examples

Many examples can be typeset from the MusicTEX distribution, but they are not included here for the sake of brevity. However we chose to produce a small type size version of

1. *Le cantique de Jean Racine* by Gabriel Fauré, in a transcription fit for organ accompanying.
2. The *Ave Maria* originally called *Méditation* by Charles Gounod, in a transcription fit for organ and voice or violin (the original is written for both a piano and an organ, which are difficult to find in the same room).
3. A part of a personal composition for the piano heavily using beams.
4. The beginning of Joseph Haydn's *aria* from the Creation, transcribed for organ and voice.

Bibliography

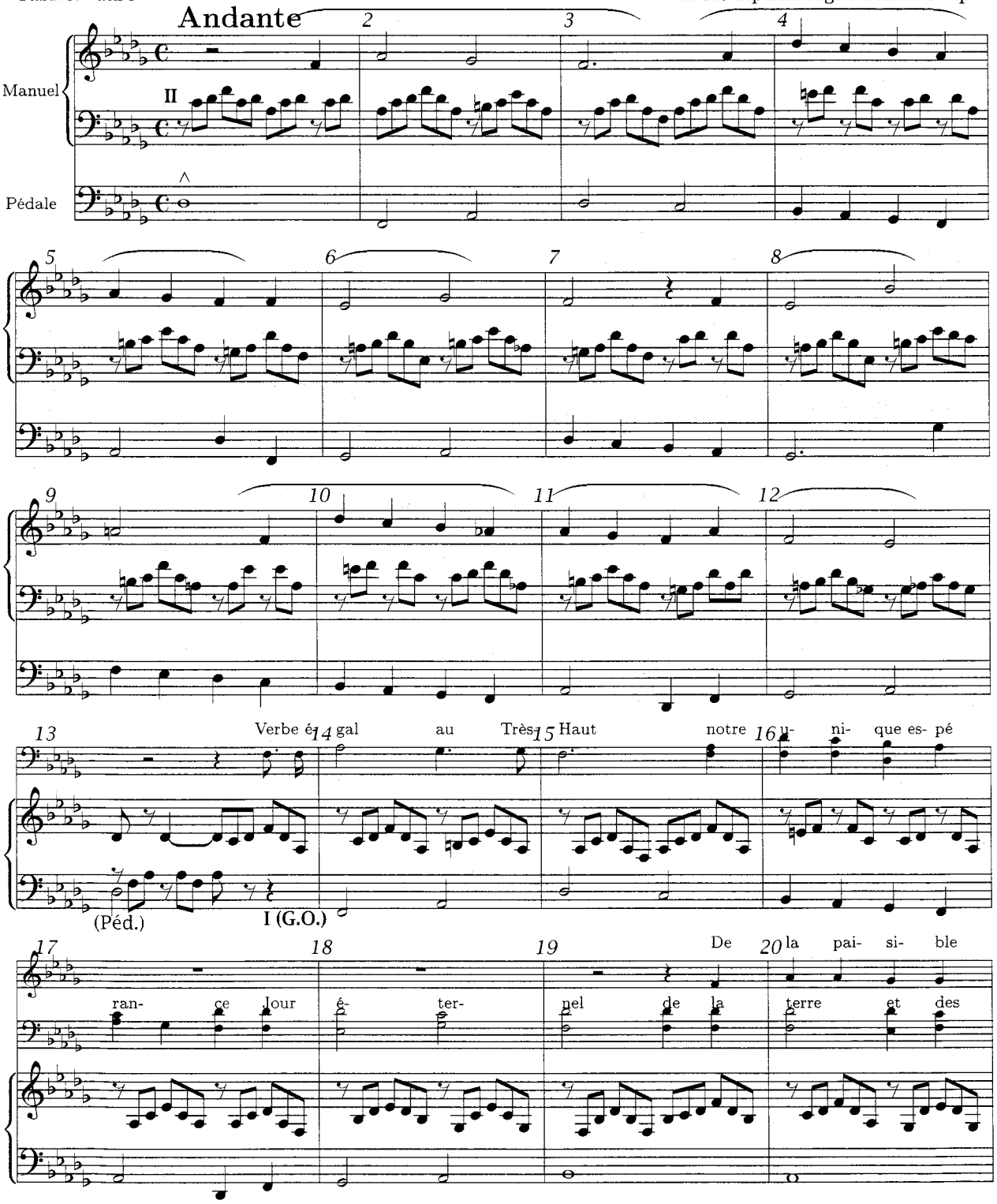
Andrea Steinbach and Angelika Schofer, *Theses* (1987, 1988), Rheinische Friedrich-Wilhelms Universität, Bonn, Germany.

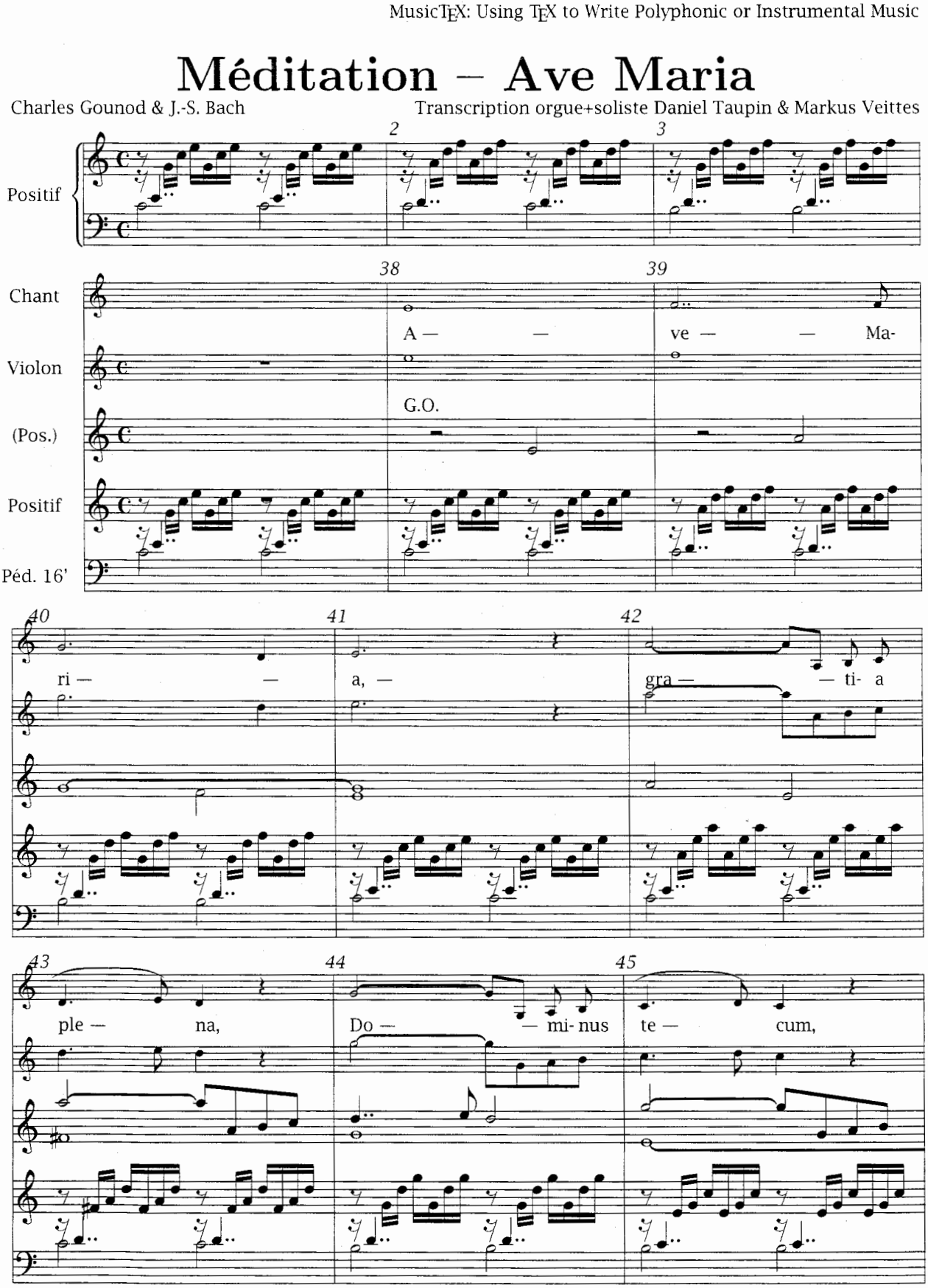
Hans Kuykens, *MIDI2TeX* (1991), available at anonymous ftp: obelix.icce.rug.nl (in directory pub/erikjan/MIDI2TeX).

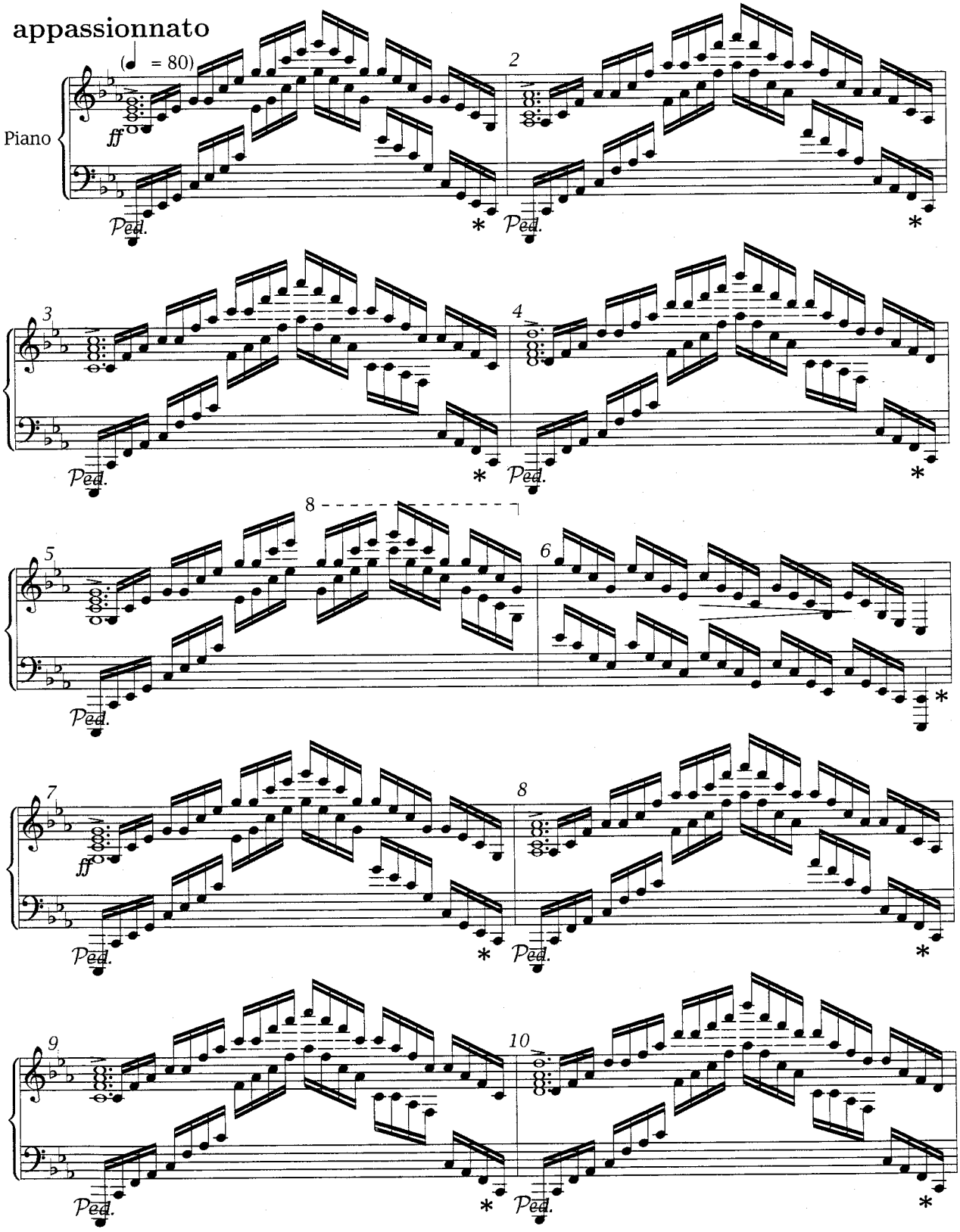
Daniel Taupin

Cantique de Jean Racine

Gabriel Fauré — Transcription orgue Daniel Taupin




Méditation – Ave Maria
Charles Gounod & J.-S. Bach
Transcription orgue+soliste Daniel Taupin & Markus Veittes
Positif
2
3
Chant
Violon
(Pos.)
Positif
Péd. 16'
38
39
A — — ve — — Ma-
G.O.
40
41
42
ri — — a, — gra — — ti- a
43
44
45
ple — na, Do — — mi- nus te — cum,


appassionnato
(♩ = 80)
Piano
ff
Ped.
*
2
3
4
8
5
6
7
8
ff
9
10

Aria No. 24
(The Creation)

Joseph HAYDN

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