MusicTEX
Using TEX to write polyphonic or instrumental music

Daniel Taupin

Laboratoire de Physique des Solides
(associé au CNRS)
bâtiment 510, Centre Universitaire,
F-91405 ORSAY Cedex

1 What is MusicTEX ?

MusicTEX is a set of TEX macros to typeset polyphonic, orchestral or polyphonic music.

Two *sizes* are available : 16pt and 20pt (standard) staff heights. For that purpose, it uses special *fonts* : `musicn16`, `slurn16`, `beamn16` and `musicn20`, `slurn20`, `beamn20` respectively.

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.

1.1 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 litterary texts to be TEX-ed. In fact, with the exception of strictly monodic instruments like most orchestral wind instruments, one should be aware that reading music actually is a matricial operation : the musician successively reads *columns* of simultaneous notes. 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
```

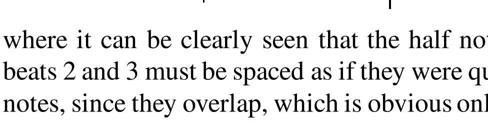
for each column of simultaneous notes.

1.2 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 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`).

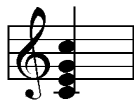
1.3 Music tokens, rather than a readymade generator

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.

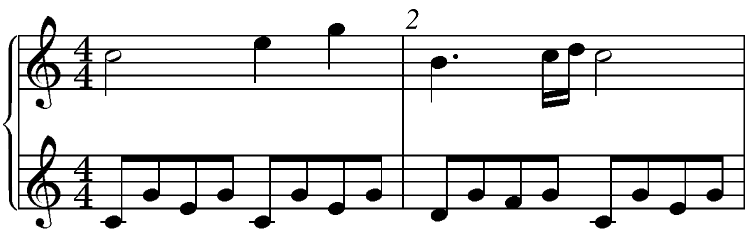


is coded `\zq c\zq e\zq g\qu j` or, in a more concise way, `\zq{ceg}\qu j` (stem up) : in fact, single notes are treated. . . like one-note chords.

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.

Besides, 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.

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



The *coding* is set as follows :

```
\def\nbinstruments{1}\relax     % a single instrument
\nbporteesi=2\relax             % with two staffs
\generalmeter{\meterfrac{4}{4}}\relax   % meter is 4/4
\debutmorceau         % initiates staffs, clefs, indication of the meter.
\normal               % normal spacing (14pt for \Notes)
\temps\Notes\ibu0f0\qh0{cge}\tbu0\qh0g|\hl j\enotes
   % \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,
   % where \qh..indicates a note hanging on a beam.
\temps\Notes\ibu0f0\qh0{cge}\tbu0\qh0g|\ql l\sk\ql n\enotes
   % \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.
```

```
\barre     % bar.
\Notes\ibu0f0\qh0{dgf}|\qlp i\enotes     % \qlp = quarter with a point.
\notes\tbu0\qh0g|\ibbl1j3\qb1j\tbl1\qb1k\enotes
   % \ibbl1j3 begins a double beam, aligned on the C (j at this pitch) of slope 0.15.
\temps\Notes\ibu0f0\qh0{cge}\tbu0\qh0g|\hl j\enotes
\suspmorceau     % closing with a simple bar (for short excerpts).
```

1.4 Other highlights

MusicTEX is intended to handle up to six instruments, each of which may be typeset on zero to four staffs (zero staff is used to record the text of vocal parts).

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, signature may change at any time as well as meters and clefs.

Besides, two frequently posed questions deserve an answer. The first one 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 not change the signature, and if – for example – you transpose by 1 pitch a piece written in C, it 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.

The second 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 this is not a serious problem for average TEX wizard apprentices.

1.5 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".

1.6 Implementation and restrictions

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 makes it doubtfully compatible with LaTEX.

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.

1.7 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.

2 Practical use

2.1 Heading statements

Before any reference to MusicTEX macros :

```
\input musicnft
\input musictex
```

which may be followed by `\input musicadd` in the case you have more than six instruments (voice is two instruments: one for the music, one for the text) or more than 6 simultaneous beams or ties or slurs.

[1] Steinbach A. & Schofer A., *Theses* (1987, 1988), Rheinische Friedrich-Wilhelms Universität, Bonn, Germany.

After that, you may write a complete book of TEX provided that you do not use `&` as a tabulation character (its `\catcode` has been changed) and that you do not overwrite MusicTEX's definitions. This means that no special macros have been designed to help you write titles, author names, comments, literature excerpts, etc.

2.2 Before you begin to write notes

You should first specify whether you want to typeset music in size 20pt par staff or 16pt. This only optional, the default value being 20pt. If you want the 16pt size, then you have to say :

```
\musicsize=16
```

Then, the first compulsory declaration is :

```
\def\nbinstruments{n}
```

where n is the number of instruments, used by MusicTEX to performs loops building staffs, setting signatures, meters, etc. Therefore, it must be defined before any other statements. An instrument may consist of several staffs, e.g. the piano. The difference between one instrument of several staffs and several instruments is as follows :

- distinct instruments may have distinct *signatures*, distinct staffs of a unique instrument share the same signature.
- *stems* may be hung to *beams* belonging to differents staffs of the same instrument.
- *chords* may extend across several staffs of the same instrument.
- staffs of a unique *instrument* are tied together with a big brace at the beginning of each line.

If the number of staffs (in french "*portées*") is not equal to one, this number must be specified by :

```
\nbporteer=p\relax
```

where p is the number of staffs, and where r is the roman numeral of the instrument considered (e.g. `\nbporteesiii` for the 3rd insrument, starting from the bottom).

If the signature is not void, one should code :

```
\generalsignature=s\relax
```

where $s > 0$ indicates the number of *sharps* in the signature and $s < 0$ the number of *flats*[2].

If there a *meter* indication is to be posted, it should be specified using the macro

```
\generalmeter{m}%
```

where m is the description of the meter indication which should appear on each staff. If it is a *fraction* (e.g. 3/4) on should code

```
\generalmeter{\meterfrac{3}{4}}%
```

or, in a simpler way (if the numbers are less than 10) :

```
\generalmeter{\meterfrac 34}%
```

Special denotations can be used, such as `\allabreve` to get ¢ and `\meterC` to get C.

However, not all music scores have the same meter in each staff. Especially, some staffs may have *ternary* meters while others have *binary*. This can be specified by using the `\generalmeter` macro to set the meter for most of the scores and overriding it by means of a more sophisticated command :

```
\metertoksii={{\meterfrac{12}8}{%
\allabreve}{}{}}%
```

which sets the meter to 12/8 for the first (lower) staff, and *alla breve* for the second staff of the instrument number 2 (*ii*). Note that there is room for 4 staffs and that void items must be specified, otherwise TEX weird errors occur.

If you want the *name of the instrument*s (or the *name of the voice*s) to be displayed in front of their respective staffs at the beginning, you may code :

```
\def\instrumentr{name of the instrument}%
```

where r is the roman numeral of the instrument considered. In this case, you should also adjust the `\parindent` dimension so that the long name of an instrument does not spill too far into the left margin.

2.3 Starting your masterpiece

Just code

```
\debutmorceau
```

which will initiate (with indentation `\parindent`) the first set of staffs for all instruments you have previously defined. But that is not sufficient to begin writing notes and silences. In fact, you also must choose the spacing of the notes. This is done by saying

```
\normal
```

if you want a elementary spacing of `10pt`. If you say `\large` the elementary spacing (`\elemskip`) is set to `12pt`. Once this is done, you can select $\sqrt{2}$ multiples of this spacing by initiating your column with `\notes` (spacing `\elemskip`), `\Notes` (spacing `1.4\elemskip`), etc. If these values of `\elemskip` are not fit to your needs, you can also say

```
\normal\elemskip=15pt
```

Of course, if your are sufficiently skilled with TEX, you can also compute `\elemskip` according to the number of bars you want in a line and the number of notes in each bar. Incidentally, this is done by the `\autolines` macro which will be introduced later.

[2] We have seen once a score in G-minor where the signature consisted in two flats (B and E) plus one sharp (F). This is not supported by MusicTEX.

In practice, the choice of the macro `\notes`, `\Notes`, `\NOtes`, etc., to initiate of column of notes sets an internal dimension register, named `\noteskip` to the given multiple of `\elemskip`. The text of the macros `\normal`, `\large`[3] and `\etroit` is thus rather simple :

```
\def\normal{\ifdim\Interligne<0.1pt%
   \computewidths\fi % (ignore this safety)
\elemskip=2\Interligne\relax %
            % (i.e. 10 points)
\def\notes{\vnotes 1.0\elemskip }%
\def\Notes{\vnotes 1.4\elemskip }%
\def\NOtes{\vnotes 2.0\elemskip }%
\def\NOTes{\vnotes 2.8\elemskip }%
\def\NOTEs{\vnotes 4.0\elemskip }%
\def\NOTES{\vnotes 5.6\elemskip }}%

\def\large{\normal\elemskip=2.4%
\Interligne}%

\def\etroit{\ifdim\Interligne<0.1pt%
\computewidths\fi
\elemskip=2\Interligne\relax
\def\notes{\vnotes 1.0\elemskip }%
\def\Notes{\vnotes 1.3\elemskip }%
\def\NOtes{\vnotes 1.8\elemskip }%
\def\NOTes{\vnotes 2.6\elemskip }%
\def\NOTEs{\vnotes 3.6\elemskip }%
\def\NOTES{\vnotes 5.2\elemskip }}%

\def\vnotes#1\elemskip{\noteskip=#1%
\elemskip\n@otes}%
```

As seen above, if these standard spacings do not meet your requirements, you are welcome to define other spacings, using the more basic macro `\vnotes`.

However, you may also prefer to ask MusicTEX to compute the size of `\elemskip` from the page width (`\hsize`) after assuming a constant number of bars of a constant number of beats within each line[4]. In this case, you have better use the `\autolines` macro, which is described below (see : "Line and page breaking").

2.4 Note pitch specification

Note pitches are usually specified by letters ranging from `a` to `z` for those which are usually written under the G-clef (`a` corresponds to the A of frequency 222.5 Hz ; the *G* of the G-clef is denoted `g`). Lower pitch notes are specified using uppercase letters ranging from `A` to `N` (the *F* of the F-clef is denoted `M`, and `F` is one octave below).

If necessary, a numeric symbol can be used to place a symbol independently of the active clef.

Besides, notes below `A` (i.e. the A of frequency 55.625 Hz), namely the lowest octave of the modern pianos can only be coded using the transposition features (see below : *transposition* and *octaviation*) or in absolute vertical position using numbers.

2.5 Writing notes

There are two major kinds of note macros :

1. those which terminate a note/chord stem and cause horizontal spacing,
2. those which initiate or extend a note/chord stem and do not cause horizontal spacing.

The first kind is used to type a melody, the second kind is used to type chords.

2.5.1 Single (spacing) notes

`\wh` p **:** whole note at pitch p.

`\hu` p **:** half note at pitch p with stem up.

`\hl` p **:** half note at pitch p with stem down.

`\qu` p **:** quarter note at pitch p with stem up.

`\ql` p **:** quarter note at pitch p with stem down.

`\cu` p **:** eighth note[5] at pitch p with stem up.

`\cl` p **:** eighth note at pitch p with stem down.

`\ccu` p **:** sixteenth note at pitch p with stem up.

`\ccl` p **:** sixteenth note at pitch p with stem down.

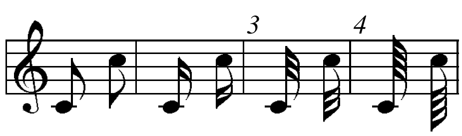
`\cccu` p **:** 32-th note at pitch p with stem up.

`\cccl` p **:** 32-th note at pitch p with stem down.

`\ccccu` p **:** 64-th note at pitch p with stem up.

`\ccccl` p **:** 64-th note at pitch p with stem down.

As an example, the sequence :



was coded as :

```
\notes\cu c\temps\cl j\enotes\barre
\notes\ccu c\temps\ccl j\enotes\barre
\notes\cccu c\temps\cccl j\enotes\barre
\notes\ccccu c\temps\ccccl j\enotes
```

If these notes are preceeded by *non-spacing* notes (i.e. macros `\zq` or `\zh`) their stem is extended up or down so as to join all notes into a single chord.

[3]To avoid problems with the LaTEX macro of the same name, this macros is only activated under LaTEX when `\begin{music}` is invoked.

[4]This does not meet the requirements of contemporaneous music, but fits very well to baroque and romantic music.

[5]The `\c` of this macro name is taken from the french word "croche" which is by the way one half of the english "crotchet" ; `\cc...`, `\ccc...` are standing for "double croche", "triple croche", etc.

2.5.2 Non-spacing (chord) notes

`\zq` p **:** quarter (or shorter) note head at pitch p with no spacing after.

`\zh` p **:** half note head at pitch p with no spacing after.

It must be pointed out that the pitch p of these notes is memorized so that the stem of the further spacing note will join them into a chord. This stem top and bottom pitch is *reset* at each spacing note.

REMARK : Notes of duration longer than whole notes are always non-spacing. This saves one useless definition, since this notes are always longer than other simultaneous ones. If needed they can be followed by `\sk` *to force spacing.*

2.5.3 Shifted non-spacing (chord) heads

These symbols are used mainly in chords where second intervals are present. It is the responsibility of the typist to choose which heads should be shifted left or right.

`\rw` p : whole note head shifted right by one note width (ca. 6pt), no spacing.
`\lw` p : whole note head shifted left by one note width (ca. 6pt), no spacing.
`\rh` p : half note head shifted right by one note width (ca. 6pt), no spacing.
`\lh` p : half note head shifted left by one note width (ca. 6pt), no spacing.
`\rq` p : quarter note head shifted right by one note width (ca. 6pt), no spacing.
`\lq` p : quarter note head shifted left by one note width (ca. 6pt), no spacing.

Except that they are shifted left of right, these macros act like `\z...` macros for stem building.

2.5.4 Shifted notes

`\rqu` **:** acts like `\qu` but the note head is shifted one note width. This is used for *chords* whose upper note has to be on the right side of the stem.

`\rhu` **:** same as above for a half note.

2.5.5 Non-spacing single notes

`\zhu` **:** half note with stem up but no spacing. It acts like `\hu` for stem building.

`\zhl` **:** half note with stem down but no spacing. It acts like `\hl` for stem building.

`\zqu` **:** quarter note with stem up but no spacing. It acts like `\qu` for stem building.

`\zql` **:** quarter note with stem down but no spacing. It acts like `\ql` for stem building.

`\lhu`, `\lhl`, `\lqu`, `\lql` **:** same as above, but the whole of the note is shifted one note width on the left.

`\zw` p **:** whole note at pitch p with no spacing after.

`\zwq` p **:** arbitrary duration note ( ) at pitch p with no spacing after.

`\zbv` p **:** breve note ( ) at pitch p with no spacing after.

`\zsb` p **:** semi-breve note ( ) at pitch p with no spacing after.

2.5.6 Pointed notes

One simple way of doing consists in putting `\pt p` to get a *dot* after the normal notehead at pitch p. Thus a quarter note with a point can be coded `\pt h\qu h`.

A simpler way of doing consists in using compact macros, namely : `\qup`, `\qupp`, `\quppp`, `\zqp`, `\zhp`, `\zwp` (these three `\z...p` are useful in chords), `\hup`, `\whp`, `\qhp`, `\qhpp`, `\qlp`, `\qlpp`, etc.

You may also introduce pointed notes, especially in groups by coding : `\qu{.a.^b.c}` which is equivalent to :

```
\pt{a}\qu{a}\pt{b}\sh{b}\qu{b}%
\pt{c}\qu{c}
```

2.6 Beams

Beams are not automatically handled, but they must be declared explicitely, *before* the first spacing note involving them is coded. Two kinds of macros are provided :

1. fixed slope beams have an arbitrary slope choosen by the user in the range -45% to +45% (by multiples of 5%) ;
2. semi-automatic beams have their slope computed knowing the number of `\noteskip` over which they are supposed to extend, and knowing the initial and final pitch of the notes they are supposed to link.

2.6.1 Fixed slope beams

`\ibu` nps **:** initiates an *upper beam* 3 horizontal line spacings above the pitch p. m is its reference number, which must be in the range [0-6] ([0-9] if `musicadd` file has been `\input`).

s is the slope of the beam. It is an integer in the range [-9,9]. $s = 1$ means a slope of 5%, $s = 9$ means a slope of 45% (the maximum with the `beamn20` or `beamn16` fonts), $s = -3$ means a slope of -15%, etc. With usual spacings a slope of 2 or 3 is fit for ascending scales. A slope of 6 to 9 is fit for ascending arpeggios.

`\ibl` nps **:** initiates a *lower beam* 3 horizontal line spacings below the pitch p. Other parameters as above.

`\ibbu` nps **:** initiates a *double upper beam* (same parameter meaning).

`\ibbl` nps **:** initiates a *double lower beam* (same parameter meaning).

`\ibbbu` nps **:** initiates a *triple upper beam* (same parameter meaning).

`\ibbbl` nps **:** initiates a *triple lower beam* (same parameter meaning).

`\ibbbbu` nps **:** initiates a *quadruple upper beam* (same parameter meaning).

`\ibbbbl` nps **:** initiates a *quadruple lower beam* (same parameter meaning).

Beam termination is also not automatic. The termination of a given beam must be explicitely declared *before* coding the last spacing note connected to that beam.

`\tbu` n **:** terminates upper beam number n at current position.

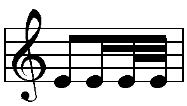
`\tbl` n **:** terminates lower beam number n at current position.

`\tbu` and `\tbl` terminate beams of any multiplicity. Therefore 32-th notes hanging on a triple beam are initiated by `\ibbbu` nps and terminated by `\tbu` n.

It is also possible to code beams whose multiplicity is not the same at the beginning. The multiplicity can be increased at any position. For instance, `\nbbu` n which sets the muliplicity of upper beam number n to 2 starting at the current position, `\nbbbu` n sets its multiciplicity to 3, and `\nbbbbu` n sets it to 4. `\nbbl` n ... `\nbbbbl` n perform the same functions for lower beams.

Notes hanging or standing on beams are coded in the form `\qh`n p and `\qb`n p where n is the beam number and p the pitch of the note head. MusicTEX adjusts the length of the note stem to link the bottom of the chord to an upper beam (normally with `\qh`) and the top of the chord to a lower beam (normally with `\qb`).

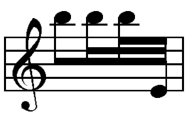
Note that the difference between upper and lower beams does not mainly consist in the beam being above or below the note heads ; rather, it specifies whether the abscissa of the beginning and the end of this beam is aligned on the right (upper beam) or on the left (lower) beam. Thus, the sequence :



has been coded as

```
\notes\ibu0e0\qh0e\nbbu0\qh0e\nbbbu0%
            \qh0e\tbu0\qh0e\enotes
```

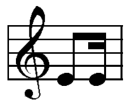
It is quite possible to terminate with `\tbu` a beam initiated with `\ibl`. This may give :



which has been coded as

```
\notes\ibl0p0\qb0p\nbbl0\qb0p\nbbbl0%
            \qb0p\tbu0\qh0e\enotes
```

Partial termination of beams is also possible, by using `\tbbu` or `\tbbl` : these macros terminate the current beam except that of order 1 (eights). `\tbbbu` or `\tbbbl` terminate the current beam except those of order 1 and 2, etc.

The macros `\tbbu` and `\tbbl` may also be invoked when only a single beam is active. Then, a second beam (upper or lower according the initiating procedure) is opened *one note width before the current position, and closed immediately*. Thus the following sequence

is coded :

```
\notes\ibu0e0\qh0e\tbbu0%
\tbu0\qh0e\enotes
```

The same behaviour occurs in the case of `\tbbbu`, `\tbbbl`, `\tbbbbu` and `\tbbbbl`.

The symmetrical pattern is also possible. For example :

has been coded as :

```
\Notes\ibbl0j0\rlap{\qsk\tbbl0}%
\qb0j\tbl0\qb0j\enotes
```

REMARK : these codings may seem complicated. In fact, it is the responsability of the user to define macros which perform the most common sequences in his masterpiece. For example, one could define sets of four sixteenths by the macro :

```
\def\qqh#1#2#3#4#5{%
\ibbl0#2#1\qh #2\qh #3\tbl0\qh #4}
```

where the first argument is the slope and the other four arguments are the pitches of the four consecutive sixteenths wanted.

2.6.2 Semi-automatic beams

In order to avoid tedious checks to adjust the slope (and even the starting pitch) of beams in music with a lot of steep beams, a set of automatically slope computing has recently been implemented (file `musicvbm.tex`). If you say `\Ibu2gj3` MusicTEX will understand that you want to build an upper beam (beam number 2) horizontally extending 3`\noteskip` and whose first note is a `g` and whose last note is `aj`. Knowing these parameters it will choose the highest slope number which corresponds to a slope not more than $(\mathtt{j}-\mathtt{g})/(3\backslash\mathtt{noteskip})$. Moreover, if there is no sufficiently steep beam slope available, then it will raise the starting point.

Eight such macros are available : `\Ibu`, `\Ibbu`, `\Ibbbu`, `\Ibbbbu`, `\Ibl`, `\Ibbl`, `\Ibbbl` and `\Ibbbbl`. Examples of their use is given in `marcello.tex`.

2.7 Phantom notes

It may be interesting, when coding a sequence of notes within a unique pair `\notes...\enotes`, to skip one note place in order – for example – to set the third note of one staff at the same abscissa as that of the second note of another staff. This can be done by inserting `\sk` which causes a spacing of one `\noteskip`[6].

If you just want to shift a note or a symbol by one note head width, you may write `\qsk`.

Note that these two latter macros must be used inside a pair `\notes...\enotes`. If you want to make a spacing of one note head width oustside, write `\nspace`.

2.8 Collective coding : sequences of notes

As seen in the MOZART example, it is not necessary to write a macro sequence `\notes...\enotes` for each column. If, on all staffs of all instruments, spacings are equal or multiple of a unique value, the notes may be concatenated in each staff : each note in each staff makes the current position horizontally advance by the elementary spacing specified by the choice of `\notes`, `\Notes`, `\NOtes`, etc.

The major interest of this feature resides in that fact that the note macros are able to write several items ; for instance `\qu{cdefghij}` writes the *C-major* scale in quarters with stem up. In the same way `\cl{abcdef^gh}` writes the *A-minor* scale in eighths. Not all note generating macros can be used to perform collective coding, but most of them can. They are :

- all the spacing notes : `\wh`, `\hu`, `\hl`... `\ccccl` and the beam hooked notes, i.e. `\qh`n and `\qb`n.
- all the chord notes whose name is of the form `\z...`

Conversely, shifted notes are not supposed to be used in collective coding, mainly because they are used in very special cases which do not deserve wasting memory space to make them collective.

If necessary a void space can be inserted in a collective coding by using `*`.

2.9 Accidentals

Accidentals can be introduced in two ways.

The first way, the *manual* way of coding them, consists for example in coding `\fl a` to put a *flat* at the pitch a, supposedly before the further note of that pitch. There is no control upon the fact that a note will be put at this position and at this pitch. Naturals, sharps, double flats and double sharps are coded `\na` p, `\sh` p, `\dfl` p and `\dsh` p respectively.

Alternate procedures `\lfl`, `\lna`, `\lsh`, `\ldfl` and `\ldsh` place the same accidentals, but their abscissa is shifted one note head width on the left. The purpose of this is to avoid collision of accidentals in a chord with narrow intervals.

The second way of coding accidentals consists in putting the symbol ^ (sharp), the symbol _ (flat), the symbol = (natural), the symbol > (double sharp), or the symbol < (double flat) within the coding of the note, e.g.: `\qh{^g}` yields a $G\sharp$. This may very well be combined with collective coding, e.g.: `\qu{ac^d}`.

Two sizes are available for accidentals. They revert to the small version when notes are supposed to be too close to each other. These two sizes can be forces by coding `\bigfl`, `\bigsh`, etc., or `\smallfl`, `\smallsh`, etc. If one does not want to have any small accidentals, then one can declare `\bigaccid` (conversely `\smallaccid` or `\varaccid` – the latter restoring variable sizes).

It is also possible to put (small) accidentals *above* the note heads. This is done using `\uppersh` p, `\upperna` p or `\upperfl` p.

2.10 Transposition and octaviation

An important feature is the existence of a special register `\transpose` whose normal value if 0. If you say

`\transpose=3`

all subsequent pitches specified by upper or lower case letters will be transposed 3 positions. If you set `\transpose` to 7 you may write your music one octave below its final pitch. Thus, you can define *octaviation* macros like

```
\def\soqu#1{\zq{#1}{\transpose=7\relax
\qu{#1}}}
```

to build quarternote octaves in a single call. Note that the octaviated note is coded within braces so that the transposition is only local.

Octaviation can also be performed in is another way, namely unsing special codes to transpose by multiples of 7 intervals. For example `\qu{'ab}` is equivalent to `\qu{hi}` and `\qu{`kl}` is equivalent to `\qu{de}`. It should be emphasized here that the `'` (*acute accent*) and the ` (*grave accent*) have cumulative ef-

[6] Do not use `\kern` nor `\hskip` : in fact `\sk` not only causes a space but also records that space for correct handling of beams.