

Suggestions for Implementing Glissando, Slur, Hammer-On, Pull-Off and Bending Articulation Improvements in MuseScore

Yves Poissant
September 11, 2020

Revision 1

Revisions History

First draft	Sept. 6, 2020	<ul style="list-style-type: none">• Throwing ideas on paper.
1 st revision	Sept. 11, 2020	<ul style="list-style-type: none">• Added subtitles and revision history.• Changed “expression” for “articulation”.• Added slanting justification in pitch functions for glissando and hammer-on / pull-off from spectrograms.• Added a GUI mock-up for editing ease-in and ease-out for glissando.• Added Glissando / Portamento MIDI stream example.• Added “most common usage” justification for proposed bend notation.• Added justification for the proposed simpler bend curve editor GUI because of the proposed notation.• Added justification for not allowing direct manipulation of bend curve but manipulate through an ease-in and ease-out properties.• Added a GUI mock-up for editing bend curves.• Added an array of example bend curves for ease-in and ease-out from 0% to 100%.• Added Bend and Hammer-On, Pull-Off example notations bend curve shaping and MIDI stream.

My Background

First some context. I don't consider myself a musician although I managed to become a lead guitarist in a rock band and had fun doing that in the 70s. I'm now a retired C++ programmer specialized in 3D photorealistic renderers who took back the guitar a few years ago to develop and improve my guitar fingerstyle technique but I like to play electric guitar riffs from time to time like the old days. And I am also acquiring music theory, which I had very sketchy notions back then.

I've been using MuseScore since about 6 years for different purposes such as choral sheet music, doing music theory homeworks, transcribing some pop and rock pieces and transcribing guitar riffs and solos. I'm quite pleased with MuseScore for all that except... for guitar solos and riffs transcriptions and playbacks rendering, which I find is missing several articulations renditions.

Thus, my proposition for implementing improvements for glissandi, slurs, hammer-ons, pull-offs and bendings, both for notations, properties editions GUIs and MIDI renditions. But keep in mind that I view those issues from an amateur guitarist perspective.

MIDI capabilities

From my research, here is my understanding of how MIDI can be used for rendering those effects. I may have missed some capabilities/technicalities though.

MIDI 1.0 already has provisions for rendering legato notes, which is required for glissando, slur, hammer-on and pull-off. This is achieved either by sending a Legato Pedal ON message or setting a legato mode. When a MIDI device is in legato mode, two notes are played legato if the first note duration overlaps the second note. In this case, the MIDI device is supposed to skip the attack portion of the second note envelope and jump right to the sustain portion.

However, those two approaches require that the instrument be played in monophony mode. While this perfectly acceptable for wind instrument, it is not acceptable for fretted instruments since it is not uncommon to play a set of legato notes on a string while holding a bass note or even a chord on the other strings.

The only other MIDI 1.0 capability available that can support those articulations without requiring that the instrument be placed in monophony mode is the pitch bend message. So for the MIDI rendering of all such articulations, pitch bending seems to be the only way to go.

A really nice way to handle most of the listed articulations would be if MIDI could allow controlling where in the sample envelope the sampler would start playing. This would allow skipping the attack and decay portions of the sample envelope by jumping from one note sample to the next note sample without restarting anew every time. Let's hope MIDI 2.0 will have this possibility.

Glissando

Notation

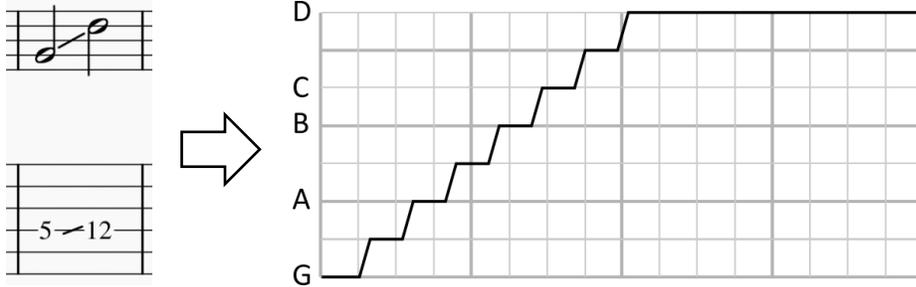
Glissando current notation is already suitable

Rendering

On a fretted instrument, glissando is usually chromatic (except when using a slide). However, a chromatic glissando on the guitar currently results in a long set of chromatic notes each with their own attack, far from the expected rendition.



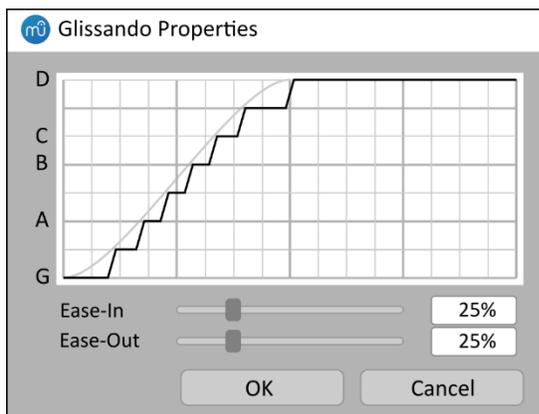
Looking at an actual guitar glissando spectrogram (on the left), glissando for fretted instrument should be rendered with a chromatic MIDI pitch bend, that is, a staircase function. The actual function is not a pure staircase though since the rising and descending edges are slightly slanted. Also, a MIDI rendition of a pure staircase function would likely introduce audible clicking artifacts.



GUI

In the above example, the glissando uses a slanted staircase function and the underlying progression curve is linear.

I'm considering exposing the underlying progression curve and letting the user modify it to model ease-in or ease-out or both as a spline curve. The staircase function would adjust accordingly. Something like the following illustration showing the Glissando properties GUI with an Ease-In and an Ease-Out of 25%:



Using MIDI pitch bend for a glissando may test the limits of the pitch bend rendition capabilities when a glissando stretches for more than a few steps, though. This might be somewhat attenuated by selecting a pivot pitch that is in the middle between the start and end pitches.

MIDI 1.0 also has a Portamento Control message but this results in a linear slide from one note to the other and is not suitable for fretted instruments.

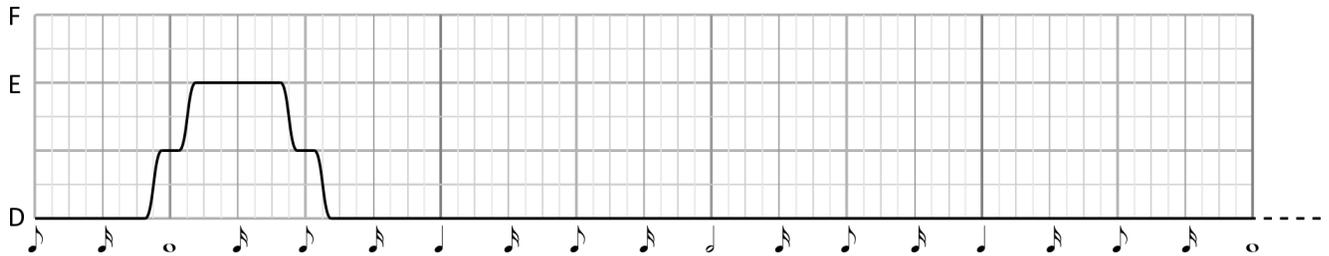
Example

Using Carlos Santana intro to Black Magic Woman measure 4 to 12 as an example. There are two (actually 4) glissandii. I'll use the glissandii between measure 6 and 7, D-E-D, with the tie on D up into measure 8.

All those four notes are concatenated together into one MIDI Bend stream of messages to avoid the attack and release portions of the guitar samples.

4 Distorted

The stream of MIDI messages models the result of the two glissandi and the tie.



Slurs, Hammer-Ons and Pull-Offs

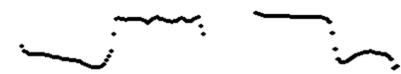
Notation

Current Slur notation in the standard staff is already suitable.

Hammer-ons and a pull-offs are usually notated simply like a slur, that is with an arc joining the two notes. This is sufficient since, on a fretted instrument, a slur cannot be performed in any other ways except with a hammer-on or a pull-off. In some books, though, an 'H' or a 'P' is placed above the slur mark, above the tabulation staff and in the middle between the two notes in the tabulation notation but not in the standard staff notation. This could be a notational option for the slur properties.

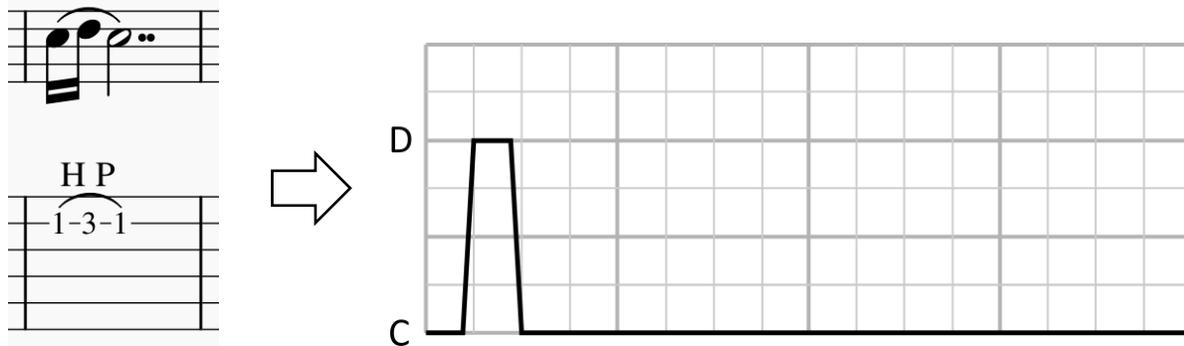
On the left, is an illustration of the two different Hammer-On and Pull-Off notations. With simple slur marks on the left and with slur marks augmented with a 'H' and a 'P' letter above the tabulation staff on the right.

Rendering



The illustration to the left shows an actual Hammer-On and Pull-Off. I'm not sure what explains the shape of the transition, but it is clearly not a straight vertical transition.

Hannel_On and Pull-Off articulations should all be rendered with a MIDI pitch bend using a smooth step function.



GUI

I don't see any needs for a Hammer-On or Pull-Off curve properties edition.

Bendings

There are too many issues with the current MuseScore bending notation and rendering to start criticizing it. I'd rather start anew.

Notation

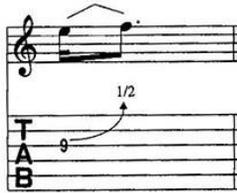
I could not find consistent bending notations between different web sites. There are at least 3 documented ways of notating bends on the web and there are many other undocumented ways. Even each of the 3 documented ways are implemented in slightly different manners by different publishers and notation softwares.

One of the documented bending notation is, by a large margin, more often used than any other though. That notation, I suggest for MuseScore, uses a pyramidal hat in the standard staff and bend curves in the tabulation staff. This has the advantage of leaving the standard staff uncluttered and places all the information relevant to the guitarist in the tabular staff. This notation also uses a small curly mark for microtone bend.

Also, using the suggested notation, the standard staff notation for bends is different from the slur notation, which allows combination of bends and slurs for precise notation without introducing confusion. Another advantage of the proposed notation is that all notes, including bended notes are explicit in the standard staff even though they are implicit (either omitted or inside parenthesis) in the tabular staff.

Here are some examples:

HALF-STEP BEND: Strike the note and bend up 1/2 step.



WHOLE-STEP BEND: Strike the note and bend up one step.



GRACE NOTE BEND: Strike the note and bend up as indicated. The first note does not take up any time.



SLIGHT (MICROTONE) BEND: Strike the note and bend up 1/4 step.



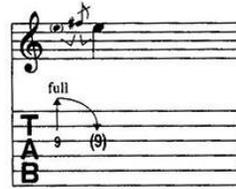
BEND AND RELEASE: Strike the note and bend up as indicated, then release back to the original note. Only the first note is struck.



PRE-BEND: Bend the note as indicated, then strike it.



PRE-BEND AND RELEASE: Bend the note as indicated. Strike it and release the bend back to the original note.

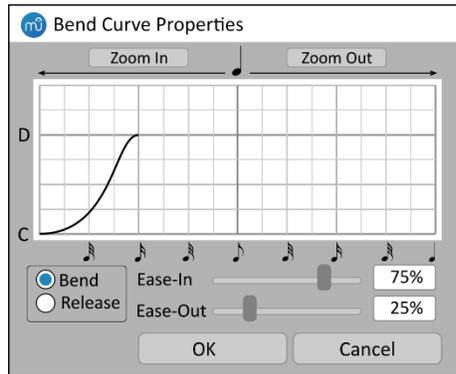


UNISON BEND: Strike the two notes simultaneously and bend the lower note up to the pitch of the higher.



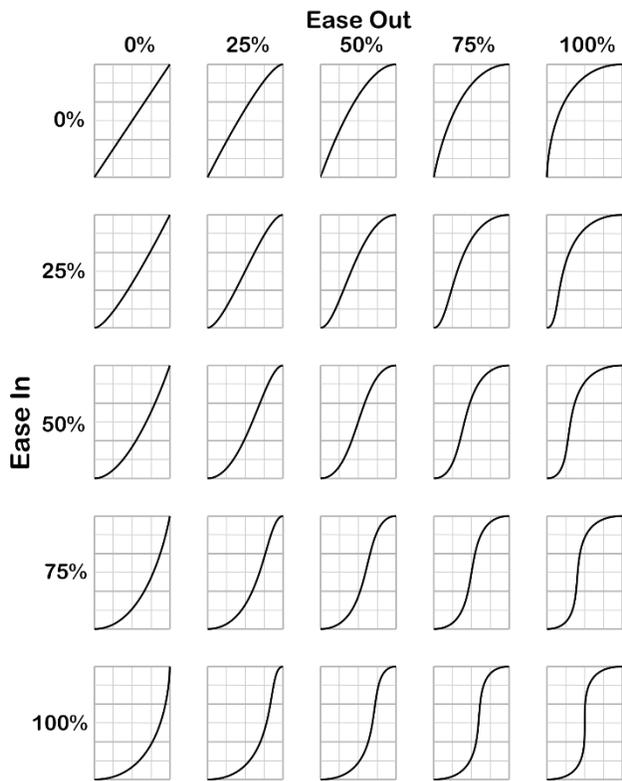
GUI

Because all the notes are explicitly notated in the standard staff, I see no needs for a complex bend curve editor GUI that allows all the current “Bend Types” such as “Bend”, “Bend / Release”, “Bend / Release / Bend”, “Pre-Bend” and “Pre-Bend / Release”. We could also add “Bend / Hold / releases”, etc.



The bend curve editor GUI only supports “Bend” and “Release” between any two consecutive notes to accommodate all possible combined sequences of bends, pre-bends, releases and holds.

The duration grid can be zoomed in or out by a factor of 2 by clicking the two buttons “Zoom-In” and “Zoom-Out”. The note span is indicated above the grid and the grid subdivision durations are indicated below the grid. The vertical subdivision is always one step and a half. A set of radio buttons allows selecting a bend curve or a release curve. The ease-in and ease-out can be set between 0% and 100% through sliders or edit boxes.



Bending curve can simulate many bending techniques including held bends.

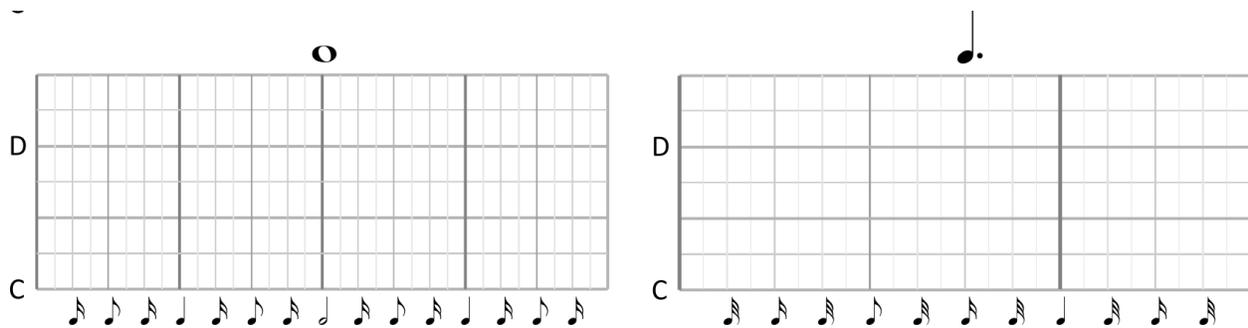
The bend curve editor needs to be able to handle much more complex bending curves than linear ones, adding an ease-in, and ease-out properties, to better simulate the inertia of a real string bend.

I suggest a bend curve property editor GUI where the curve is modeled with an Ease-In and Ease-Out properties. When each of those properties are set at 0%, there are no ease-in nor ease-out and the bend curve is a straight linear function. Both ease-in and ease-out properties can be set at values between 0% and 100% to model different bend shapes.

The illustration to the left shows all combinations of ease-in and ease-out for values between 0% and 100% at every 25% steps for a guitar bend. The curves must be vertically flipped for bend releases.

The end user cannot directly control the shape of the bending curve as would be the case in a graphics application. The curve shape is strictly controlled by the two ease-in and ease-out properties to avoid illegal time-dependent transfer curves.

The bend curve editor horizontal grid must adjust to the note duration such that it is easy to set bend curve control points at exact sixteen, eight, quarter, half, etc. positions with snap functionality and two control points may be positioned one above the other for sharp step function modeling.



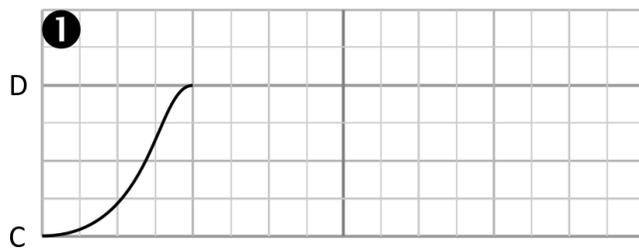
Bend rendering already uses the MIDI pitch bend functionality.

Example bend settings for a score

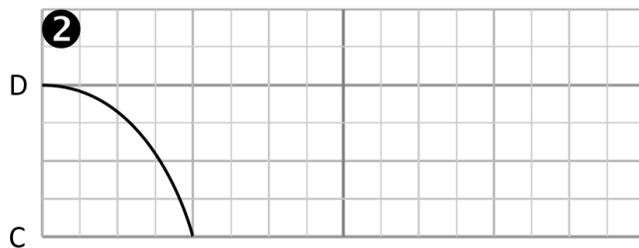
The example uses a part of Carlos Santana "Black Magic Woman" guitar intro. The score is captured from Musescore and modified in Inkscape to add the proposed notation. I used regular 16th notes for bend starts because grace note playback is broken in Musescore (*the 16th grace note is played with the correct duration but the following note is delayed by half its duration, leaving a large rest between the*

grace note and the following note¹). But IMO, grace notes should be used there. Of course, the numbered bullets would not show in a score.

First I will detail each numbered bullets bend settings individually. After I described them all, I will show how this is translated to MIDI messages. The bend curves are adjusted so they match as close as possible the pitch bends I observe in the spectrograms.

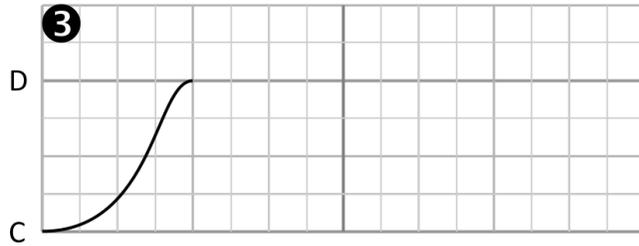


Zoom spans a 4th
 Duration is a 16th
 Bend type is "Bend"
 Ease-In is 75%
 Ease-Out is 25%

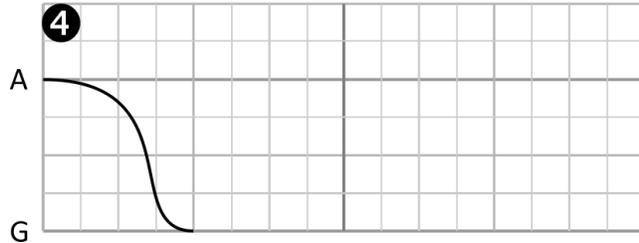


Zoom spans a 4th
 Duration is a 16th
 Bend type is "Release"
 Ease-In is 75%
 Ease-Out is 0%

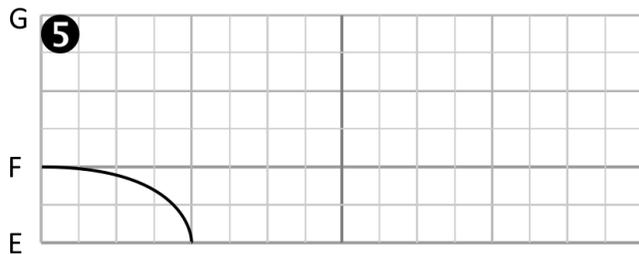
¹ I'm not sure if I should file an issue report about that because I read arguments on the forum that this is the correct behavior. This sure feels very strange to my ears though.



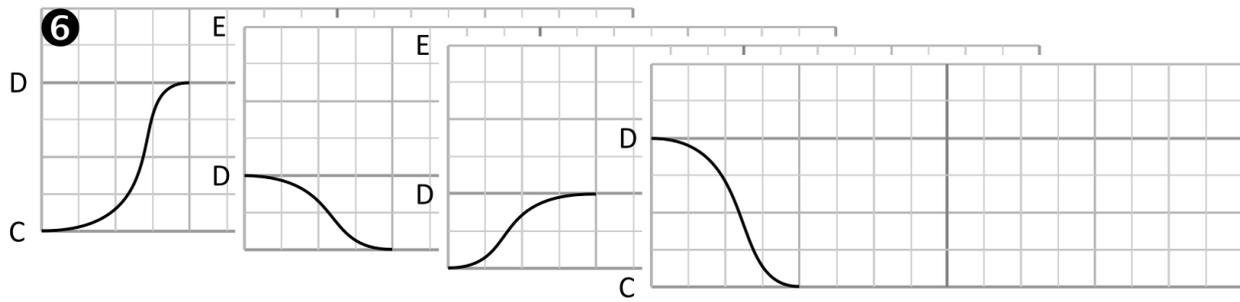
Zoom spans a 4th
 Duration is a 16th
 Bend type is "Bend"
 Ease-In is 75%
 Ease-Out is 25%



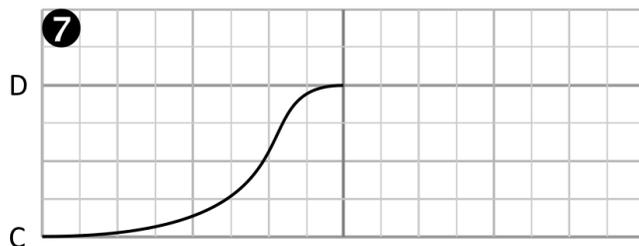
Zoom spans a 4th
 Duration is a 16th
 Bend type is "Release"
 Ease-In is 100%
 Ease-Out is 50%



Zoom spans a 4th
 Duration is a 16th
 Bend type is "Release"
 Ease-In is 100%
 Ease-Out is 0%



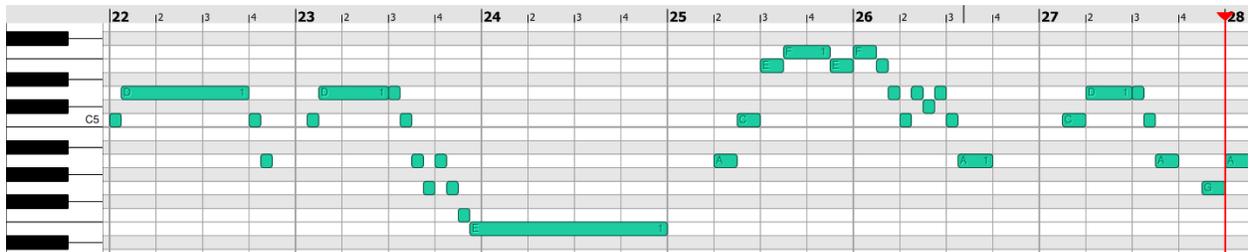
Zooms span a 4th
 Durations are a 16th
 Bend types are "Bend", "Release", "Bend", "Release"
 Ease-Ins are 100%, 75%, 50%, 75%
 Ease-Outs are 50%, 50%, 75%, 50%



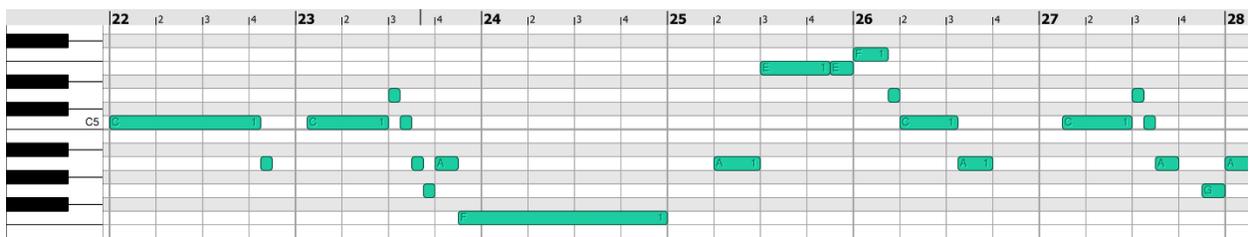
Zoom spans a 4th
 Duration is an 8th
 Bend type is "Bend"
 Ease-In is 100%
 Ease-Out is 38%

MIDI Rendition

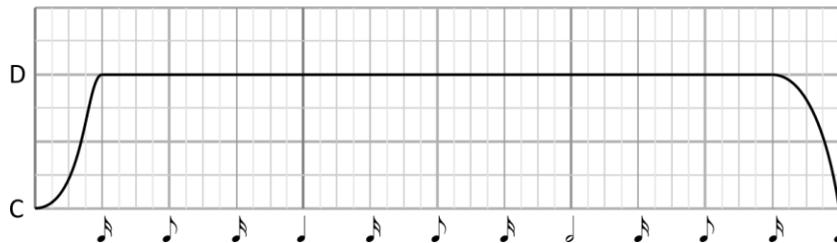
When rendering, notes that are linked through a tie, a slur or a bend are concatenated together and sent as a series of bend messages. Without this concatenation, the MIDI stream would look like this:



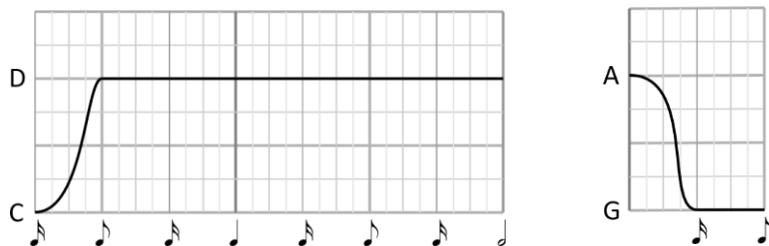
With the concatenation, the MIDI stream looks like this:



The stream of MIDI messages to model the Bend, Hold and Release in measure 22 looks like this:



The stream of MIDI messages to model the Bend and Hold in measure 23.1 to 23.2 and the Pre-Bend and Release in measure 23.4 looks like these:



The stream of MIDI messages to model the Pre-Bend, Release and Hold from measure 23.4 and 24 looks like this:

