# TONES Automation

anematode

January 2, 2019

## 1 Who is this intended for?

It's intended for me. I might read this in a few months because I didn't know what the hell I was doing. Thus, it's written in a somewhat understandable way, but it won't try to explain ideas that aren't a product of this project.

## 2 What is Automation?

Automation should be an abstraction of the mapping from a beat number, $b$, to a value of a certain parameter, $p$. In other words, it tells us what the value of a parameter is at some point in beat space (not in *time*, per se, because that is tempo-dependent.) This is very important in the course of modulating parameters to go with the music, and therefore this part of TONES must be well made, fast, sufficiently generalized, and amenable to easy conversion into Web Audio API format. There will definitely be a disconnect between the mathematical description of automation and its actual implementation, but the former should serve as a helpful guidance.

## 3 The Distilled Automation Model

Define a function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}$, which takes in the time in beats since the start of the automation and outputs the value of the parameter at that beat. $f$ need not be continuous; parameters should be able to change instantaneously (in Web Audio, this will correspond to an instant change between two samples). We model this function as a series of $n$ disjoint (apart from their endpoints) segments $s_1, s_2, \cdots, s_n$ covering $[0, l]$, where $l$ is the length of the automation. The boundaries of these segments are $x_1, x_2, \cdots, x_{n+1} \in \mathbb{R}_{\geq 0}$, so the boundaries of segment $s_i$ are $x_i$ and $x_{i+1}$. Note that $x_1 = 0$ and $x_{n+1} = l$. We should think of these segments as functions; the domain of $s_i$ is $[x_i, x_{i+1}]$, and we define $f$ as the piecewise

$$f(b) = \begin{cases} s_1(b), x_1 \le b < x_2 \\ s_2(b), x_2 \le b < x_3 \\ \cdots \\ s_n(b), x_n \le b < x_{n+1} \\ s_n(x_{n+1}), x_{n+1} \le b \end{cases} .$$

Note that we use $\le$ for the lower bound and $<$ for the upper bound. This is important in the context of instantaneous changes, which are modeled as segments of length 0. Furthermore, all $b$ greater than the last separator, $x_{n+1}$, have $f(b) = s_n(x_{n+1})$; in essence, the last otherwise-defined value of $f$. The complete description of $f$ is given by the separators $x_i$, the functions $s_i$, and the number of segments $n$; the separators effectively restrict the domains of $s_i$.

Derived properties of $f(b)$, which will prove useful:

$Dom(f) = [0, \infty)$

$l := x_{n+1}$, the length of the automation

$pmax := \max(f)$, the maximum value $f$ attains (more precisely/correctly, the supremum of the image of $f$, because there are cases where $pmax \neq f(a)$ for any $a$ due to a discontinuity)

$pmin := \min(f)$, the minimum value $f$ attains (more precisely/correctly, the infimum of the image of $f$, because there are cases where $pmin \neq f(a)$ for any $a$ due to a discontinuity)

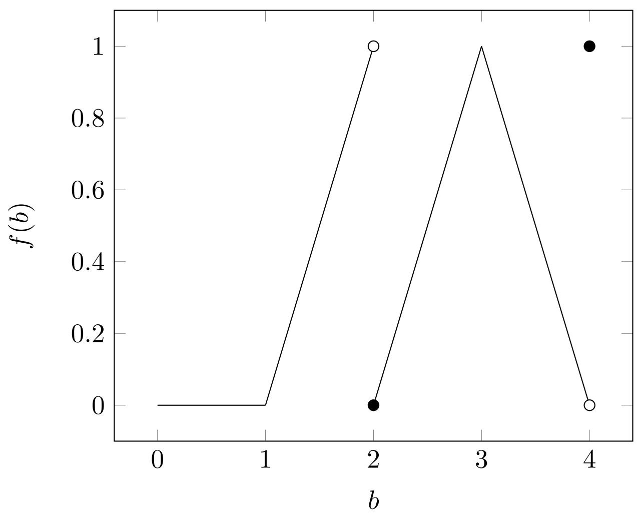
$Rng(f) \subseteq [pmin, pmax]$. Note that the range can be equal to this interval, or the same interval with open boundaries, or even with gaps in the middle (due to discontinuities).

$derivative(x) := f'(x)$, the derivative of $f$, with domain $Dom(derivative) \subseteq [0, \inf) \backslash \{x_i\}$. Note that $f'(x)$ is undefined if and only if $x \in x_i$ (and $f$ is non-differentiable at $x_i$ because $1 < i \le n \wedge \lim_{x \to x_i^+} s'_{i-1}(x) \neq \lim_{x \to x_i^-} s'_i(x)$ or $i = n+1 \wedge \lim_{x \to x_{n+1}^+} s'_n(x) \neq 0$)

$integral(x) := \int_0^x f(x)dx$, the definite integral of $f$ from 0 to $x$, with domain $[0, \inf)$. Because $f$ is defined for all $x \ge 0$, this function should be defined for all $x \ge 0$ as well by the Fundamental Theorem of Calculus

$timeIntegral(x) := \int_0^x \frac{1}{f(x)}dx$, the definite integral of $1/f$ from 0 to $x$. This integral should exist iff $pmin > 0$, otherwise it's nonsensical for its intended purpose: tempo automations.

Let's see how we'd define an actual automation that highlights some of these properties of $f$.



Obviously, automations will also have non-linear segments, but this is for simplicity. We have $l = 4$, $n = 5$, $x_i = 0, 1, 2, 3, 4, 4$, and

$$s_i = \{0, x - 1, x - 2, -x + 4, 1\}.$$

Here are some values of $f$, the derivative, and the integral, spaced evenly from 0 to 5:

$$f(0) = s_1(0) = 0; integral(0) = 0; derivative(0) = 0$$

[Note that the derivative is defined at 0 because $\lim_{x\to 0^+} f(x) = 0$]

$$f(.5) = s_1(.5) = 0; integral(.5) = 0; derivative(.5) = 0$$

$$f(1) = s_2(1) = 0; integral(1) = 0; derivative(1) = \text{undef}$$

[$f$ is continuous, but non-differentiable, at 1]

$$f(1.5) = s_2(1.5) = 0.5; integral(1.5) = 0.125; derivative(1.5) = 1$$

$$f(2) = s_3(2) = 0; integral(2) = 0.5; derivative(2) = \text{undef}$$

$$f(2.5) = s_3(2.5) = 0.5; integral(2.5) = 0.625; derivative(2.5) = 1$$

$$f(3) = s_4(3) = 1; integral(3) = 1; derivative(3) = \text{undef}$$

$$f(3.5) = s_4(3.5) = 0.5; integral(3.5) = 1.375; derivative(3.5) = -1$$

$$f(4) = s_5(4) = 1; integral(4) = 1.5; derivative(4) = \text{undef}$$

$$f(4.5) = s_5(4) = 1; integral(4.5) = 2; derivative(4.5) = 0$$

$$f(5) = s_5(4) = 1; integral(5) = 2.5; derivative(5) = 0.$$

For the case where an instantaneous change occurs in the middle of an automation, note that in this case we would have two consecutive and equal $x_i$:

$x_j = x_{j+1}$ for $j < n$. Then, the corresponding segment will have no $x$ values corresponding to it in the piecewise, including $x = x_j$! That's because the domain of $s_j$ will be $x_j \leq x < x_{j+1}$, which has no solutions; instead $x = x_j$ will fall in segment $s_{j+1}$ because that has domain $x_{j+1} \leq x < x_{j+2}$. This is desirable behavior: instantaneous changes will have no effect in the middle of an automation, but they will signal a change in a parameter right at the end of an automation if necessary. This is useful for resetting parameters quickly, for example.

In summary, we can define an mapping $f : \mathbb{R}_{\geq 0} \to \mathbb{R}$ taking in a beat time $b$ and spitting out the value of a parameter at that time, uniquely defined by $n > 0$ (number of segments), $s_1, ..., s_n$ (segment functions), and $x_1, \cdots, x_{n+1}$ (domain separators), with certain consequent properties. The rest of this document is about the actual implementation.

## 4 AutomationSegment

A parent, abstract class which all the segment functions will be based on.

Properties:

$x_1$: starting x value ($\geq 0$)

$x_2$: ending x value ($\geq x_1$)

$y_1$: starting y value

$y_2$: ending y value

Functions:

constructor($x_1$, $y_1$, $x_2$, $y_2$): class constructor, rejecting $x_2 < x_1$

length(): returns $x_2 - x_1$

deltaY(): returns $y_2 - y_1$

valueAt(x): returns getValues(x), taking in a single value

derivativeAt(x): returns getDerivatives(x), taking in a single value

integralAt(x): returns getIntegrals(x), taking in a single value

timeIntegralAt(x): returns getTimeIntegrals(x), taking in a single value

Children should implement:

ymin(): minimum of the segment over the range $[x_1, x_2]$.

ymax(): maximum of the segment over the range $[x_1, x_2]$.

**Note: the following functions should take in Float64Arrays and evaluate the specified quantity for each array entry, then rewrite that array entry.**

getValues(x): returning the value of the segment at beat x. Only has to be valid for the domain $[x_1, x_2]$, but it can be continued to other values if that makes things easier.

getDerivatives(x): returning the value of the derivative at beat x. Similarly, only has to be valid for $[x_1, x_2]$.

getIntegrals(x): returning the value of

$$\int_{x_1}^{x} s(x)dx;$$

only has to be valid for $[x_1, x_2]$.

getTimeIntegrals(x): returning the value of

$$\int_{x_1}^{x} \frac{1}{s(x)} dx;$$

only has to be valid for $[x_1, x_2]$ and if ymin() $> 0$.

**End Note.**

translateX(x): translate the segment left or right by x units and return itself

translateY(y): translate the segment up or down by y units and return itself

scaleX(x): scale the segment **relative to 0** in the x-axis and return itself

scaleY(y): scale the segment **relative to 0** in the y-axis and return itself

clone(): clone the segment

## 5 ConstantAutomationSegment

Child of Automation Segment.

A constant between $x_1, c$ and $x_2, c$.

Properties (besides parent's):

c: the value of the constant parameter

Functions:

constructor($x_1$, $x_2$, c): calls super($x_1$, c, $x_2$, c)

ymin: c

ymax: c

getValues(x): c

getDerivatives(x): 0

getIntegrals(x): $c(x - x_1)$

getTimeIntegrals(x): $(x - x_1)/c$

translateX(x): $x_1 := x + x_1$, $x_2 := x + x_2$

translateY(y): $y_1 := y + y_1$, $y_2 := y + y_2$

scaleX(x): $x_1 := xx_1$, $x_2 := xx_2$; flip $(x_1, y_1)$ and $(x_2, y_2)$ if necessary

scaleY(y): $y_1 := yy_1$, $y_2 := yy_2$

clone(): simply construct the segment with parameters $x_1, y_1, x_2, y_2$.

## 6 LinearAutomationSegment

Child of Automation Segment.

Simple linear interpolation between $(x_1, y_1)$ and $(x_2, y_2)$, which also works for constant values if $y_1 = y_2$.

No extra properties.

Functions:

constructor(...args): rejects $x_2 = x_1$, then calls super(...args)

ymin: $\min\{y_1, y_2\}$

ymax: $\max\{y_1, y_2\}$

getValues(x):

$$(x - x_1) \cdot \frac{y_2 - y_1}{x_2 - x_1} + y_1$$

getDerivatives(x):

$$\frac{y_2 - y_1}{x_2 - x_1}$$

Derivation:

$$\frac{d}{dx}\left((x - x_1) \cdot \frac{y_2 - y_1}{x_2 - x_1} + y_1\right) = \frac{y_2 - y_1}{x_2 - x_1}$$

getIntegrals(x):

$$y_1(x - x_1) + \frac{m(x - x_1)^2}{2},$$

where $m = \frac{y_2-y_1}{x_2-x_1}$ is the slope.

Derivation: If $m = \frac{y_2-y_1}{x_2-x_1}$, then

$$getIntegrals(x) = \int_{x_1}^{x} (t - x_1) \cdot m + y_1 dt = \int_0^{x-x_1} mt + y_1 dt.$$

The antiderivative is $bt + \frac{mt^2}{2}$, so the value of the definite integral is

$$y_1 t + \frac{mt^2}{2}\Big|_{t=0}^{t=x-x_1} = \underbrace{y_1(x - x_1)}_{\text{rectangle}} + \underbrace{\frac{m(x - x_1)^2}{2}}_{\text{triangle}}$$

The underbraces are for a sanity check.

getTimeIntegrals(x)=

$$\begin{cases} \frac{\log(m(x-x_1)+y_1)-\log(y_1)}{m}, m \neq 0 \\ \frac{x-x_1}{y_1}, m = 0 \end{cases}$$

Derivation: We have $\int \frac{1}{mx+b}dx = \frac{\log(mx+b)}{m} + C$ if $m \neq 0$, where log is ln. Of course, we have $b = y_1$ and $m = \frac{y_2-y_1}{x_2-x_1}$, so we have

$$\int_{x_1}^{x} \frac{1}{s(t)}dt = \int_0^{x-x_1} \frac{1}{s(t + x_1)}dt = \frac{\log(mt + y_1)}{m}\Big|_{t=0}^{t=x-x_1} = \frac{\log(m(x - x_1) + y_1) - \log(y_1)}{m}.$$

If $m = 0$, then we're just integrating $1/y_1$ from $x_1$ to $x$, giving $\frac{x-x_1}{y_1}$. In practice, it's probably intelligent to choose some arbitrary $0 < \epsilon \ll 1$ and use this second formula for $|m| < \epsilon$, because dividing by very small $m$ is not good float-wise. [TODO] figure out a better way to calculate getTimeIntegrals for very, very small $|m|$.

translateX(x): $x_1 := x + x_1$, $x_2 := x + x_2$

translateY(y): $y_1 := y + y_1$, $y_2 := y + y_2$

scaleX(x): reject $x = 0$; $x_1 := xx_1$, $x_2 := xx_2$; flip $(x_1, y_1)$ and $(x_2, y_2)$ if necessary

scaleY(y): $y_1 := yy_1$, $y_2 := yy_2$

clone(): simply construct the segment with parameters $x_1, y_1, x_2, y_2$.

## 7 ExponentialAutomationSegment

Child of Automation Segment.

To define an exponential automation segment, we take $x_1, y_1, x_2, y_2$, but also a y-value $y_c$ so that

$$s(x_1) = y_1$$
$$s(x_2) = y_2$$
$$s\left(\frac{x_1 + x_2}{2}\right) = y_c$$

Note that when $y_c = \frac{y_1+y_2}{2}$, we have a plain old linear interpolation. We want $s$ to have the form

$$s(x) = ae^{bx} + c.$$

Since this is monotonically increasing or decreasing, we have ymin() $< y_c <$ ymax().

There are numerous strategies we can use here to efficientify algorithms, which we will soon explore.

Properties (besides parent's):

$y_c$: the aforementioned variable that determines the curvature of the exponential segment.

Let's figure how to determine $a, b, c$ from the five variables. First, let's translate everything so that $(x_1, y_1) = (0, 0)$ and $(x_2, y_2) = (1, 1)$; at the end we will translate everything back. We have $x_1' = 0$, $y_1' = 0$, $x_2' = 1$, $y_2' = 1$, and $y_c' = (y_c - y_1)/(y_2 - y_1)$. In this translated system, we want to find $a', b', c'$.

We have that

$$a'e^{b'(0)} + c' = 0 \Longrightarrow c' = -a'$$

$$a'e^{b'} + c' = 1 \Longrightarrow a'e^{b'} - a' = 1$$

$$a'e^{b'/2} + c' = y_c' \Longrightarrow a'e^{b'/2} - a' = y_c'$$

Dividing the last two right hand equations we get

$$\frac{e^{b'} - 1}{e^{b'/2} - 1} = \frac{1}{y_c'}$$

$$\frac{(e^{b'/2} + 1)(e^{b'/2} - 1)}{e^{b'/2} - 1} = \frac{1}{y_c'}$$

$$e^{b'/2} + 1 = \frac{1}{y_c'}$$

$$e^{b'/2} = \frac{1}{y'_c} - 1$$

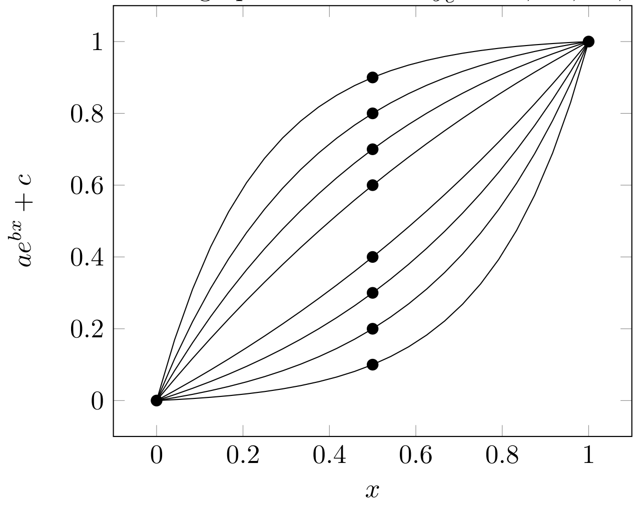
$$b'/2 = \log\left(\frac{1}{y'_c} - 1\right)$$

$$b' = 2\log\left(\frac{1}{y'_c} - 1\right)$$

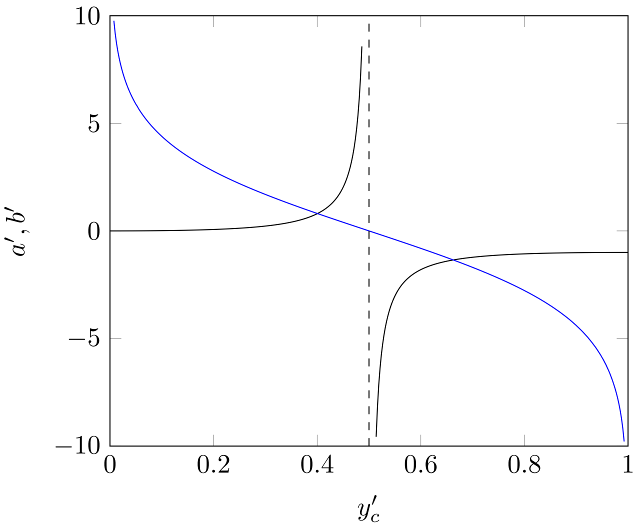
Since $a'(e^{b'}-1) = 1$, we have $a' = \frac{1}{e^{b'}-1} = \frac{1}{\left(\frac{1}{y'_c}-1\right)^2-1}$. $c'$ is just the negation of this.

Before we translate this back, we need to take note of (1) what this looks like and (2) what to do in the edge cases (there's a very important one here!).

This is what the graph looks like for $y'_c = 0.1, 0.2, 0.3, 0.4, 0.6, 0.7, 0.8, 0.9$:



Note that 0.5 is conspicuously missing: we'll see why now. We have $0 < y'_c < 1$; here is the graph of $a'$ (black) and $b'$ (blue) for those values of $y'_c$:



This isn't good: $a'$ gets far too big for some acceptable $y'_c$ values! $b'$ also gets too big for $y'_c$ very close to 0 or 1, but this doesn't really matter because it grows much slower. We should figure out a way to compute $a'e^{b'x} + c'$ more easily.

Since $c' = -a'$, we have

$$a'e^{b'x} + c' = a'(e^{b'x} - 1).$$

Some succulent algebra shows that this equals

$$\frac{y_c'^2}{1 - 2y'_c}\left(\left(\frac{1}{y'_c} - 1\right)^{2x} - 1\right).$$

Intuition dictates that the limit of this equation as $y'_c \to 0.5$ is $x$. Indeed, it is a 0/0 indeterminate form, so applying L'Hopital's rule we get

$$\lim_{y'_c \to 1/2} \frac{\left(y_c'^2\left(\left(\frac{1}{y'_c} - 1\right)^{2x} - 1\right)\right)'}{(1 - 2y'_c)'}$$

$$= \lim_{y'_c \to 1/2} \frac{2y'_c\left(\left(\frac{1}{y'_c} - 1\right)^{2x} - 1\right) + y_c'^2\left(-2x\left(\frac{1}{y'_c} - 1\right)^{2x-1} y_c'^{-2}\right)}{-2}$$

$$= \frac{2 \cdot .5\left(\left(\frac{1}{.5} - 1\right)^{2x} - 1\right) + .5^2\left(-2x\left(\frac{1}{.5} - 1\right)^{2x-1} .5^{-2}\right)}{-2}$$

$$= \frac{2 \cdot .5\left((1)^{2x} - 1\right) + .5^2\left(-2x\,(1)^{2x-1}\, .5^{-2}\right)}{-2}$$

$$= \frac{0 - .5^{2-2}2x}{-2} = \frac{-2x}{-2} = x.$$

So should we just default to $x$ if $y'_c$ is sufficiently close to 0.5 so that floats become explosive? This is, sadly, the method I ended up using. [TODO] Figure out a better calculation for ExponentialAutomationSegment for $y_c$ near 0.5.

Define $\epsilon = 10^{-9}$. If $y'_c$ is within $\epsilon$ of 0.5, we use a linear interpolation.

Now let's transform back to our original space. To do this, we have $x$ turn into $(x - x_1)/(x_2 - x_1)$ and transform the entire expression $y$ by $y(y_2 - y_1) + y_1$. Finally, we have $y'_c = (y_c - y_1)/(y_2 - y_1)$. Written out, this looks like

$$\frac{y_c'^2}{1 - 2y'_c}\left(\left(\frac{1}{y'_c} - 1\right)^{\frac{2(x-x_1)}{x_2-x_1}} - 1\right)(y_2 - y_1) + y_1.$$

It may be a good idea to precompute many of these constants. In any case, let us continue the specification.

constructor(x1, y1, x2, y2, yc): calls super(x1, y1, x2, y2), rejects $x2 = x1$

ymin: $\min\{y_1, y_2\}$

ymax: $\max\{y_1, y_2\}$

getValues(x): As discussed, it is

$$\frac{y_c'^2}{1 - 2y'_c}\left(\left(\frac{1}{y'_c} - 1\right)^{\frac{2(x-x_1)}{x_2-x_1}} - 1\right)(y_2 - y_1) + y_1$$

where $y'_c = (y_c - y_1)/(y_2 - y_1)$ if $|y'_c - 0.5| > \epsilon$, otherwise it is

$$(x - x_1)\frac{y_2 - y_1}{x_2 - x_1} + y_1.$$

getDerivatives(x):

$$\left(\frac{y_c'^2}{1 - 2y'_c}\left(\left(\frac{1}{y'_c} - 1\right)^{\frac{2(x-x_1)}{x_2-x_1}} - 1\right)(y_2 - y_1) + y_1\right)'$$

$$= \frac{y_c'^2}{1 - 2y'_c}(y_2 - y_1)\left(\left(\frac{1}{y'_c} - 1\right)^{\frac{2(x-x_1)}{x_2-x_1}}\right)'$$

$$= \boxed{\frac{y_c'^2}{1 - 2y'_c}(y_2 - y_1)\left(\frac{2}{x_2 - x_1}\right)\log\left(\frac{1}{y'_c} - 1\right)\left(\frac{1}{y'_c} - 1\right)^{\frac{2(x-x_1)}{x_2-x_1}}}$$

if $|y'_c - 0.5| > \epsilon$, otherwise it is

$$\frac{y_2 - y_1}{x_2 - x_1}.$$

getIntegrals(x):

$$\int_{x_1}^{x} \frac{y_c'^2}{1-2y_c'}\left(\left(\frac{1}{y_c'}-1\right)^{\frac{2(t-x_1)}{x_2-x_1}}-1\right)(y_2-y_1)+y_1 dt$$

$$=\int_{0}^{x-x_1} \frac{y_c'^2}{1-2y_c'}\left(\left(\frac{1}{y_c'}-1\right)^{\frac{2t}{x_2-x_1}}-1\right)(y_2-y_1)+y_1 dt$$

$$=\int_0^{x-x_1} y_1 dt+\frac{y_c'^2}{1-2y_c'}(y_2-y_1)\int_0^{x-x_1}\left(\frac{1}{y_c'}-1\right)^{\frac{2t}{x_2-x_1}}-1dt$$

$$=y_1(x-x_1)+\frac{y_c'^2}{1-2y_c'}(y_2-y_1)\left(x_1-x+\int_0^{x-x_1}\left(\frac{1}{y_c'}-1\right)^{\frac{2t}{x_2-x_1}}dt\right)$$

$$=y_1(x-x_1)+\frac{y_c'^2}{1-2y_c'}(y_2-y_1)\left(x_1-x+\int_0^{x-x_1}\left(\frac{1}{y_c'}-1\right)^{\frac{2t}{x_2-x_1}}dt\right)$$

$$=y_1(x-x_1)+\frac{y_c'^2}{1-2y_c'}(y_2-y_1)\left(x_1-x+\left(\frac{\left(\left(\frac{1}{y_c'}-1\right)^{\frac{2}{x_2-x_1}}\right)^t}{\log\left(\left(\frac{1}{y_c'}-1\right)^{\frac{2}{x_2-x_1}}\right)}\Bigg|_{t=0}^{t=x-x_1}\right)\right)$$

Thus, let $v=\left(\frac{1}{y_c'}-1\right)^{\frac{2}{x_2-x_1}}$. Then getIntegrals(x) is

$$\boxed{y_1(x-x_1)+\frac{y_c'^2}{1-2y_c'}(y_2-y_1)\left(x_1-x+\frac{v^{x-x_1}-1}{\log(v)}\right)} \qquad (1)$$

if $|y_c'-0.5|>\epsilon$, otherwise it is

$$y_1(x-x_1)+\frac{m(x-x_1)^2}{2},$$

where

$$m=\frac{y_2-y_1}{x_2-x_1}.$$

Alternatively, (1) is

$$\boxed{y_1(x-x_1)+\frac{y_c'^2}{1-2y_c'}(y_2-y_1)\left(x_1-x+\frac{\left(\left(\frac{1}{y_c'}-1\right)^{\frac{2(x-x_1)}{x_2-x_1}}-1\right)(x_2-x_1)}{2\log\left(\frac{1}{y_c'}-1\right)}\right)}$$

getTimeIntegrals(x):

This won't be pleasant to evaluate. To do it more easily, let's define some constants to substitute back later.

$$\int_{x_1}^{x} \frac{1}{\frac{y_c'^2}{1-2y_c'}\left(\left(\frac{1}{y_c'}-1\right)^{\frac{2(t-x_1)}{x_2-x_1}}-1\right)(y_2-y_1)+y_1}dt$$

$$= \int_{0}^{x-x_1} \frac{1}{\frac{y_c'^2}{1-2y_c'}\left(\left(\frac{1}{y_c'}-1\right)^{\frac{2t}{x_2-x_1}}-1\right)(y_2-y_1)+y_1}dt$$

Let $a = \frac{y_c'^2}{1-2y_c'}(y_2-y_1)$, $v = \left(\frac{1}{y_c'}-1\right)^{\frac{2}{x_2-x_1}}$, and $b = y_1$. Then our integral is just

$$\int_0^{x-x_1} \frac{1}{a(v^t-1)+b}dt$$

$$= \int_0^{x-x_1} \frac{1}{av^t-a+b}dt$$

$$= \left.\frac{\log(a(v^t-1)+b)-t\log v}{(a-b)\log v}\right|_{t=0}^{t=x-x_1}$$

$$= \boxed{\frac{\log(a(v^{x-x_1}-1)+b)-(x-x_1)\log v-\log(b)}{(a-b)\log v}}$$

if $|y_c'-0.5| > \epsilon$, otherwise we define it to be

$$\begin{cases} \frac{\log(m(x-x_1)+y_1)-\log(y_1)}{m}, m \neq 0 \\ \frac{x-x_1}{y_1}, m = 0 \end{cases}$$

where

$$m = \frac{y_2-y_1}{x_2-x_1}.$$

translateX(x): $x_1 := x + x_1$, $x_2 := x + x_2$

translateY(y): $y_1 := y + y_1$, $y_2 := y + y_2$, $y_c := y + y_c$

scaleX(x): $x_1 := xx_1$, $x_2 := xx_2$; flip $(x_1, y_1)$ and $(x_2, y_2)$ if necessary

scaleY(y): $y_1 := yy_1$, $y_2 := yy_2$, $y_c = yy_c$

clone(): simply construct the segment with parameters $x_1, y_1, x_2, y_2, y_c$.

## 8 ParabolicAutomationSegment

Child of AutomationSegment.

This is very similar to the Exponential case, but we use a parabola to interpolate between $(x_1, y_1), ((x_1 + x_2)/2, y_c), (x_2, y_2)$. It should have the form $s(x) = ax^2 + bx + c$.

Properties (besides parent's):

$y_c$: variable determining the curvature of the parabolic segment.

For the getValues function we can use the Lagrange Interpolation Formula for three variables, which tells us that

$$s(x) = \frac{y_1(x - x_2)(x - m)}{(x_1 - x_2)(x_1 - m)} + \frac{y_2(x - x_1)(x - m)}{(x_2 - x_1)(x_2 - m)} + \frac{y_c(x - x_1)(x - x_2)}{(m - x_1)(m - x_2)}$$

where $m = (x_1 + x_2)/2$.

$$s(x) = \frac{y_1(x - x_2)(x - m)}{(x_1 - x_2)^2/2} + \frac{y_2(x - x_1)(x - m)}{(x_2 - x_1)^2/2} + \frac{y_c(x - x_1)(x - x_2)}{-(x_2 - x_1)^2/4}$$

$$= \frac{2y_1(x - x_2)(x - m)}{(x_1 - x_2)^2} + \frac{2y_2(x - x_1)(x - m)}{(x_2 - x_1)^2} - \frac{4y_c(x - x_1)(x - x_2)}{(x_2 - x_1)^2}$$

$$= \frac{2y_1(x - x_2)(x - m) + 2y_2(x - x_1)(x - m) - 4y_c(x - x_1)(x - x_2)}{(x_1 - x_2)^2}$$

$$= \frac{2(x - m)(y_1(x - x_2) + y_2(x - x_1)) - 4y_c(x - x_1)(x - x_2)}{(x_1 - x_2)^2}.$$

For ymin and ymax, we'll need to know the vertex of this parabola, so let's do the getDerivatives first, which should just be a linear function of $x$.

$$\left(\frac{2(x - m)(y_1(x - x_2) + y_2(x - x_1)) - 4y_c(x - x_1)(x - x_2)}{(x_1 - x_2)^2}\right)'$$

$$= (x_1-x_2)^{-2}(4x_2y_c(x-x_1)-4xy_c(x-x_1)-2my_2(x-x_1)-2my_1(x-x_2)+2xy_2(x-x_1)+2xy_1(x-x_2))'$$

$$= (x_1-x_2)^{-2}((4y_1+4y_2-8y_c)x+4x_2y_c+4x_1y_c-2my_2-2my_1-2x_1y_2-2y_1x_2).$$

When this equals 0, we have the vertex at

$$x = -\frac{4x_2y_c + 4x_1y_c - 2my_2 - 2my_1 - 2x_1y_2 - 2y_1x_2}{4y_1 + 4y_2 - 8y_c}$$

Functions:

constructor($x_1$, $y_1$, $x_2$, $y_2$, $y_c$): calls super($x_1$, $y_1$, $x_2$, $y_2$), rejects $x_2 = x_1$

Let $m = (x_1 + x_2)/2$, $n = 4x_2y_c + 4x_1y_c - 2my_2 - 2my_1 - 2x_1y_2 - 2y_1x_2$, and $d = 4y_1 + 4y_2 - 8y_c$.

ymin:

$$= \begin{cases} \min\{y_1, y_2\}, d \leq 0 \\ \min\{y_1, y_2\}, d \neq 0 \wedge \neg(x_1 < n/d < x_2) \\ getValues(n/d), otherwise \end{cases}$$

ymax:

$$= \begin{cases} \max\{y_1, y_2\}, d \geq 0 \\ \max\{y_1, y_2\}, d \neq 0 \wedge \neg(x_1 < n/d < x_2) \\ getValues(n/d), otherwise \end{cases}$$

getValues(x):

$$\frac{2(x-m)(y_1(x-x_2)+y_2(x-x_1)) - 4y_c(x-x_1)(x-x_2)}{(x_1-x_2)^2}$$

getDerivatives(x):

$$(x_1-x_2)^{-2}((4y_1+4y_2-8y_c)x + 4x_2y_c + 4x_1y_c - 2my_2 - 2my_1 - 2x_1y_2 - 2y_1x_2)$$

getIntegrals(x):

$$\int_{x_1}^{x} \frac{2(x-m)(y_1(x-x_2)+y_2(x-x_1)) - 4y_c(x-x_1)(x-x_2)}{(x_1-x_2)^2} dx$$

$$= (x_1-x_2)^{-2} \int_{x_1}^{x} 4x_2y_cx - 4x_2y_cx_1 - 4xy_cx + 4xy_cx_1 - 2my_1x + 2my_1x_2 \\ - 2mxy_2 + 2mx_1y_2 + 2x^2y_2 - 2xx_1y_2 + 2xy_1x - 2xy_1x_2 dx$$

$$= (x_1-x_2)^{-2} \int_{x_1}^{x} (2y_1+2y_2-4y_c)x^2 + (4x_2y_c+4y_cx_1-2my_1-2my_2-2x_1y_2-2y_1x_2)x \\ + (-4x_2y_cx_1 + 2my_1x_2 + 2mx_1y_2)dx$$

$$= (x_1-x_2)^{-2}\Bigg(\frac{(2y_1+2y_2-4y_c)x^3}{3} + \frac{(4x_2y_c+4y_cx_1-2my_1-2my_2-2x_1y_2-2y_1x_2)x^2}{2} \\ + (-4x_2y_cx_1 + 2my_1x_2 + 2mx_1y_2)x\Big|_{x=x_1}^{x=x}\Bigg)$$

$$= (x_1 - x_2)^{-2}\left(\frac{\overbrace{(2y_1 + 2y_2 - 4y_c)}^{=d/2}(x^3 - x_1^3)}{3} + \frac{\overbrace{(4x_2y_c + 4y_cx_1 - 2my_1 - 2my_2 - 2x_1y_2 - 2y_1x_2)}^{=n}(x^2 - x_1^2)}{2} + (-4x_2y_cx_1 + 2my_1x_2 + 2mx_1y_2)(x - x_1)\right)$$

getTimeIntegrals(x): Once again, not pleasant. We know that we can express the segment as a quadratic $s(x) = ax^2 + bx + c$, and unlike the exponential segment, there's no degenerate case when there's a line besides the fact that $a = 0$. We can figure out $a, b, c$ later; let's integrate

$$\int_{x_1}^{x} \frac{dt}{at^2 + bt + c}.$$

One should note that this is not defined for all lower and upper bounds for the definite integral, because we may integrate over an asymptote, but this *does* converge if $ymin() > 0$. Let's find the antiderivative.

Completing the square and using the tangent antiderivative gives

$$\int \frac{1}{ax^2 + bx + c}dx = \frac{2}{\sqrt{-q}} \tan^{-1}\left(\frac{2ax + b}{\sqrt{-q}}\right),$$

where $q = b^2 - 4ac$ is the discriminant, so the definite integral is

$$\frac{2}{\sqrt{-q}}\left(\tan^{-1}\left(\frac{2ax + b}{\sqrt{-q}}\right) - \tan^{-1}\left(\frac{2ax_1 + b}{\sqrt{-q}}\right)\right)$$

To find $a, b, c$ we just expand out the formula for getValues and compare terms:

$$(x_1 - x_2)^{-2}(\underbrace{(2y_1 + 2y_2 - 4y_c)}_{a}x^2 + \underbrace{(4x_2y_c + 4y_cx_1 - 2my_1 - 2my_2 - 2x_1y_2 - 2y_1x_2)}_{b}x + \underbrace{(-4x_2y_cx_1 + 2my_1x_2 + 2mx_1y_2)}_{c})$$

$$a = \frac{2y_1 + 2y_2 - 4y_c}{(x_1 - x_2)^2}$$
$$b = \frac{4x_2y_c + 4y_cx_1 - 2my_1 - 2my_2 - 2x_1y_2 - 2y_1x_2}{(x_1 - x_2)^2}$$
$$c = \frac{-4x_2y_cx_1 + 2my_1x_2 + 2mx_1y_2}{(x_1 - x_2)^2}$$

Great! But what if $q = 0$ or $q > 0$? In the former case, we'd be dividing by 0; in the latter case, we'd be involving imaginary numbers. Let's deal with each separately.

If $q = 0$ and $a \neq 0$, then the original integral is just

$$\frac{1}{a}\int_{x_1}^{x} \frac{dt}{\left(t + \frac{b}{2a}\right)^2} = \frac{1}{a}\left( -\frac{1}{t + b/2a}\Big|_{t=x_1}^{t=x} \right)$$
$$= -\frac{1}{a}\left( \frac{1}{x + b/2a} + \frac{1}{x_1 + b/2a} \right)$$
$$= -\left( \frac{2}{2ax + b} + \frac{2}{2ax_1 + b} \right)$$

If $a = 0$ and $b \neq 0$, then the original integral is just

$$\int_{x_1}^{x} \frac{1}{bt + c} = \frac{\log(bx + c) - \log(bx_1 + c)}{b},$$

and if $a = b = 0$ then we get

$$\frac{x - x_1}{c}.$$

If $q > 0$, then $\sqrt{-q} = i\sqrt{q}$; the latter square root should allow us to compute without introducing imaginary numbers.

$$\frac{2}{i\sqrt{q}}\left(\tan^{-1}\left(\frac{2ax + b}{i\sqrt{q}}\right) - \tan^{-1}\left(\frac{2ax_1 + b}{i\sqrt{q}}\right)\right).$$

We have that $\tan^{-1}(x/i) = \tan^{-1} -ix = i/2(\log(1 - ix) - \log(1 + ix)) = i/2(\log(1 - x) - \log(1 + x)) = -i\tanh^{-1} x$ :

$$\frac{2}{i\sqrt{q}} i\left(-\tanh^{-1}\left(\frac{2ax + b}{\sqrt{q}}\right) + \tanh^{-1}\left(\frac{2ax_1 + b}{\sqrt{q}}\right)\right)$$
$$\frac{2}{\sqrt{q}}\left(-\tanh^{-1}\left(\frac{2ax + b}{\sqrt{q}}\right) + \tanh^{-1}\left(\frac{2ax_1 + b}{\sqrt{q}}\right)\right).$$

In the end, we have the following getTimeIntegrals(x):

$$getTimeIntegrals(x) = \begin{cases} \frac{x - x_1}{c}, a = b = 0 \\ \frac{\log(bx+c) - \log(bx_1+c)}{b}, a = 0 \wedge b \neq 0 \\ -\left(\frac{2}{2ax+b} + \frac{2}{2ax_1+b}\right), q = 0 \wedge a \neq 0 \\ \frac{2}{\sqrt{-q}}\left(\tan^{-1}\left(\frac{2ax+b}{\sqrt{-q}}\right) - \tan^{-1}\left(\frac{2ax_1+b}{\sqrt{-q}}\right)\right), q < 0 \wedge a \neq 0 \\ \frac{2}{\sqrt{q}}\left(-\tanh^{-1}\left(\frac{2ax+b}{\sqrt{q}}\right) + \tanh^{-1}\left(\frac{2ax_1+b}{\sqrt{q}}\right)\right), q > 0 \wedge a \neq 0 \end{cases}$$

translateX(x): $x_1 := x + x_1$, $x_2 := x + x_2$
translateY(y): $y_1 := y + y_1$, $y_2 := y + y_2$, $y_c := y + y_c$
scaleX(x): $x_1 := xx_1$, $x_2 := xx_2$; flip $(x_1, y_1)$ and $(x_2, y_2)$ if necessary
scaleY(y): $y_1 := yy_1$, $y_2 := yy_2$, $y_c := yy_c$
clone(): simply construct the segment with parameters $x_1, y_1, x_2, y_2, y_c$.

## 9 Automation

Now it's time to combine these segments into an actual automation. We store each segment in an array, and we require that the first segment start at $x = 0$. Furthermore, we need the endpoints of each segment to line up correctly.

Properties:

segments, the array of segments (possibly empty)

Functions:

constructor(segments = []): takes in an array of segments; initializing with no segments is allowed

ymin(): Infinity or

$$\min\{s_i.ymin()\}$$

ymax(): -Infinity or

$$\max\{s_i.ymax()\}$$

get segCount(): segments.length

get length(): segCount() > 0 ? segments[segments.length - 1].x2 : 0

getSegment(i): segments[i], throws error if out of bounds

setSegment(i, seg): sets segments[i] to seg

insertSegment(i, seg): inserts a segment at i

removeSegment(i): removes a segment at i

removeSegmentIf(func): removes a segment if func evaluates to true

clearSegments(): removes all segments

isConsistent(): checks whether the segments are consistent in terms of boundaries

addSegment(seg): appends the segment to the end of the automation, shifting it by the necessary amount

shiftSegments(i, units): shift all segments after and including i to the right or left by units

getValues(arr): takes in an array, fills it up with the values defined by the piecewise

getDerivatives(arr): takes in an array, fills it up with the derivatives defined by the piecewise

getIntegrals(arr): takes in an array, fills it up with the integrals defined by the piecewise

getTimeIntegrals(arr): takes in an array, fills it up with the time integrals defined by the piecewise

valueAt(c): value of curve at c, internally calling getValues

derivativeAt(c): derivative of curve at c, internally calling getDerivatives

integralAt(c): integral of curve at c, internally calling getIntegrals

timeIntegralAt(c): timeIntegral of curve at c, internally calling getTimeIntegrals

_timeIntegralCheck(): checks whether ymin() ¿ 0

scaleX(x): scale the automation by a scale factor, x, flipping it over if necessary

scaleY(y): scale the automation vertically by a scale factor y