HYPERCURVE

Johann Philippe, Jacopo Greco d'Alceo

A hybrid curve forge in Csound

Introduction

HYPERCURVE is a laboratory for curves shaping. New library designed to combine different curve algorithms inside one function table, it has been thought as a tool for musicians looking to shape precisely their envelopes and function tables.

Available as Csound plugin opcodes, as well as C++, Lua or Faust library.

Why HYPERCURVE

- Composing piecewises hybrid curves with several algorithms
- Manipulating curves with specific functions
- *Proper and universal syntax* > unlike GEN, we can directly combine curve algorithms
- A *library* of curves: a place where existing can be found and used
- A *laboratory* for curves: a place where we can imagine/experiment new approaches and new algorithms (mathematically, musically, etc.)

Why HYPERCURVE

As far as today we can appreciate the basic waveforms of a synthesizer, in further researches, we could imagine the recognition of some specific and peculiar curves in a pedagogical interest (e.g. a music theory of curves) and in artistic and musical composition.

EXEMPLE

- Amplitude
- Frequency
- Rhythm
- Filtre and others

gi1 → 3segments Tightrope walker curve

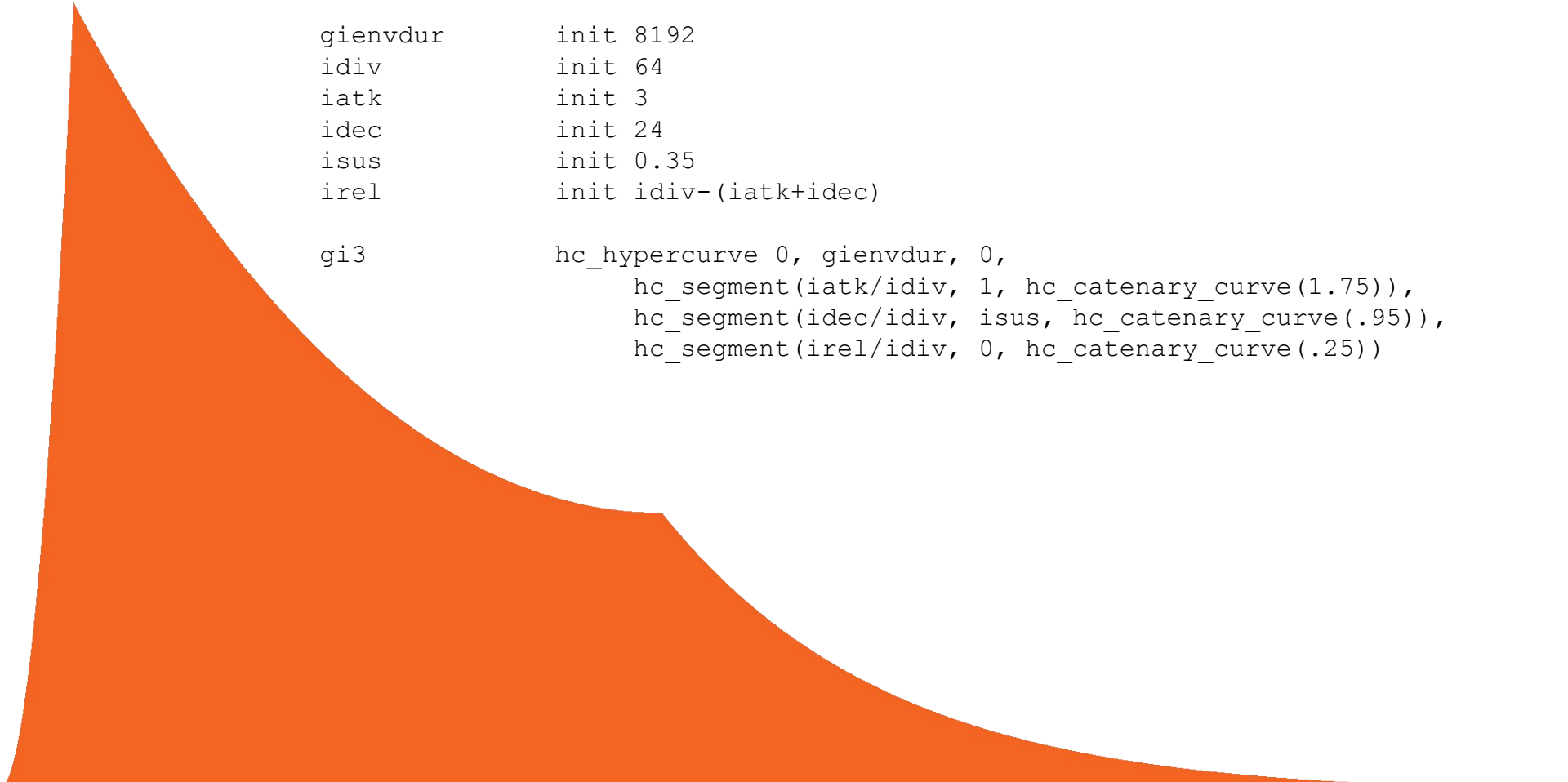
```
gienvdur        init 8192
idiv            init 64
iatk            init 3
idec            init 24
isus            init 0.35
irel            init idiv-(iatk+idec)

gi1             hc_hypercurve 0, gienvdur, 0,
                     hc_segment(iatk/idiv, 1, hc_tightrope_walker_curve(1.105, .125)),
                     hc_segment(idec/idiv, isus, hc_tightrope_walker_curve(.95, .25)),
                     hc_segment(irel/idiv, 0, hc_tightrope_walker_curve(.5, .15))
```

gi2 → 3segments Mouse or Kiss curve

```
gienvdur        init 8192
idiv            init 64
iatk            init 3
idec            init 24
isus            init 0.35
irel            init idiv-(iatk+idec)

gi2             hc_hypercurve 0, gienvdur, 0,
                     hc_segment(iatk/idiv, 1, hc_kiss_curve()),
                     hc_segment(idec/idiv, isus, hc_kiss_curve()),
                     hc_segment(irel/idiv, 0, hc_kiss_curve())
```

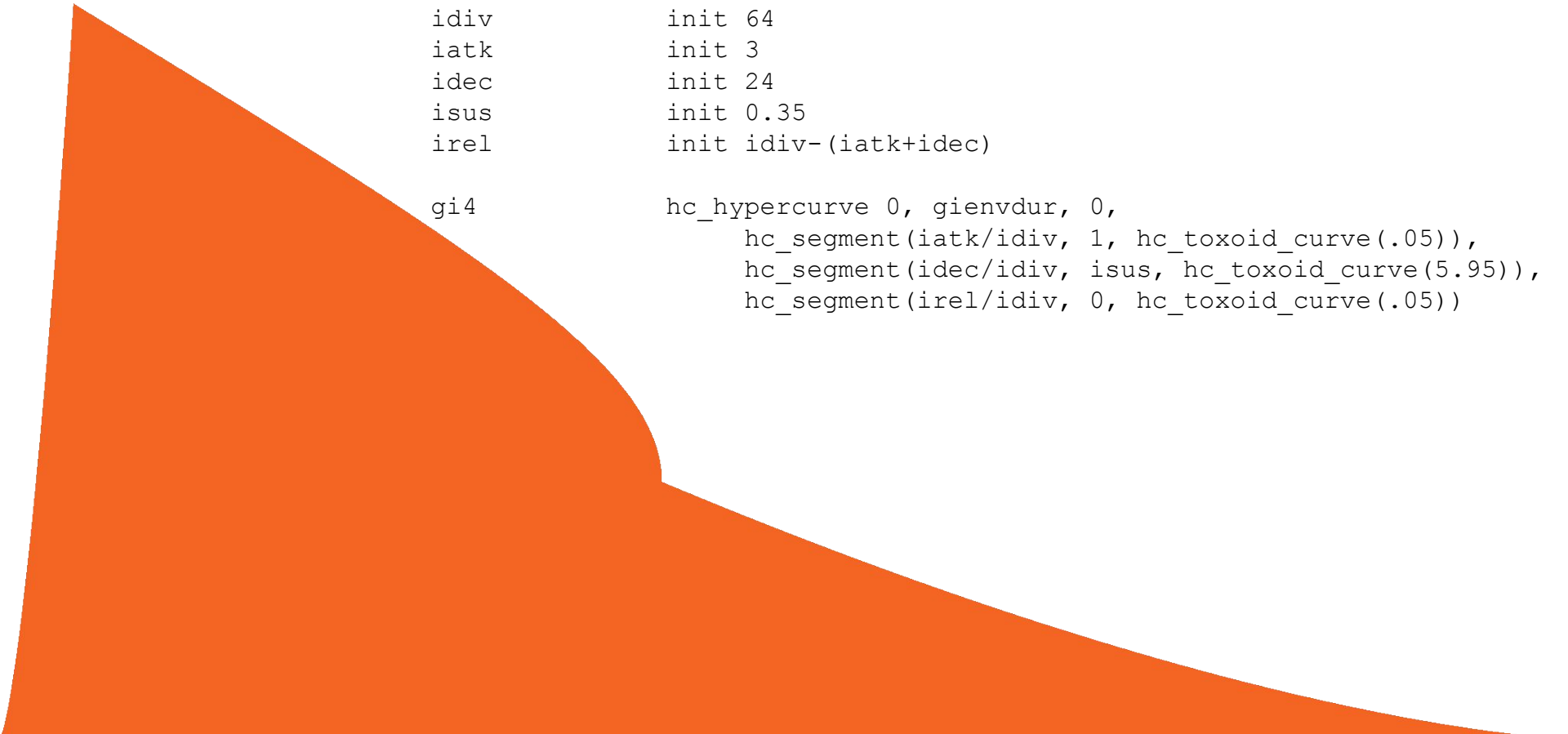
gi3 → 3segments Catenary curve

```
gienvdur        init 8192
idiv            init 64
iatk            init 3
idec            init 24
isus            init 0.35
irel            init idiv-(iatk+idec)

gi3             hc_hypercurve 0, gienvdur, 0,
                     hc_segment(iatk/idiv, 1, hc_catenary_curve(1.75)),
                     hc_segment(idec/idiv, isus, hc_catenary_curve(.95)),
                     hc_segment(irel/idiv, 0, hc_catenary_curve(.25))
```



gi4 → 3segments Toxoid curve

```
gienvdur        init 8192
idiv            init 64
iatk            init 3
idec            init 24
isus            init 0.35
irel            init idiv-(iatk+idec)

gi4             hc_hypercurve 0, gienvdur, 0,
                     hc_segment(iatk/idiv, 1, hc_toxoid_curve(.05)),
                     hc_segment(idec/idiv, isus, hc_toxoid_curve(5.95)),
                     hc_segment(irel/idiv, 0, hc_toxoid_curve(.05))
```

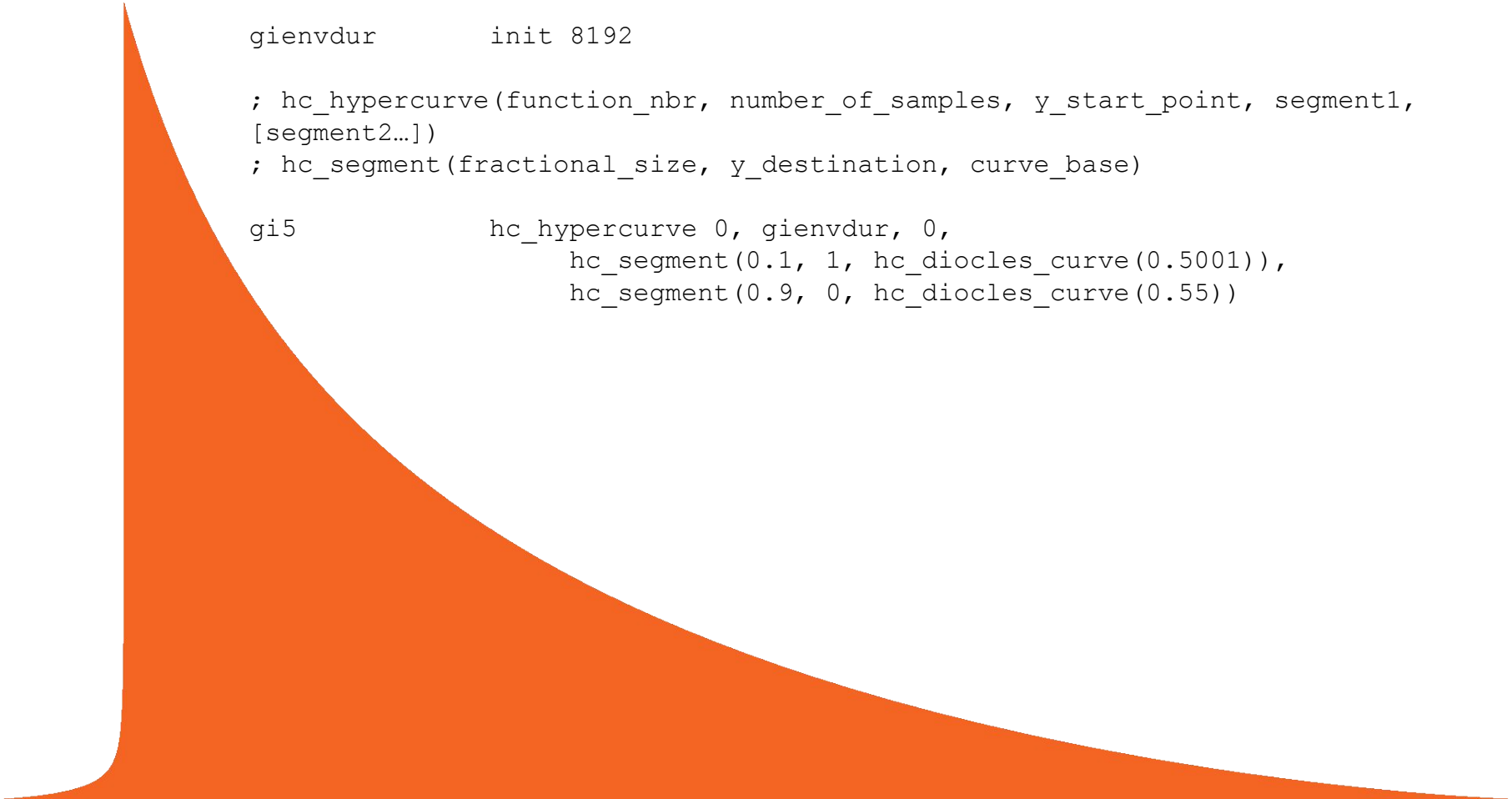


Basic principles of HYPERCURVE

- How to get it : github.com/johannphilippe/hypercurve
- Syntax : 3 main components :
    - hc_hypercurve
    - hc_segment
    - Curve algorithms (aka curve base)
- Manipulations (operators, mirror, invert, concatenate, resize, normalize)
- Implementation in Csound as function tables : readable with table, oscil (...)

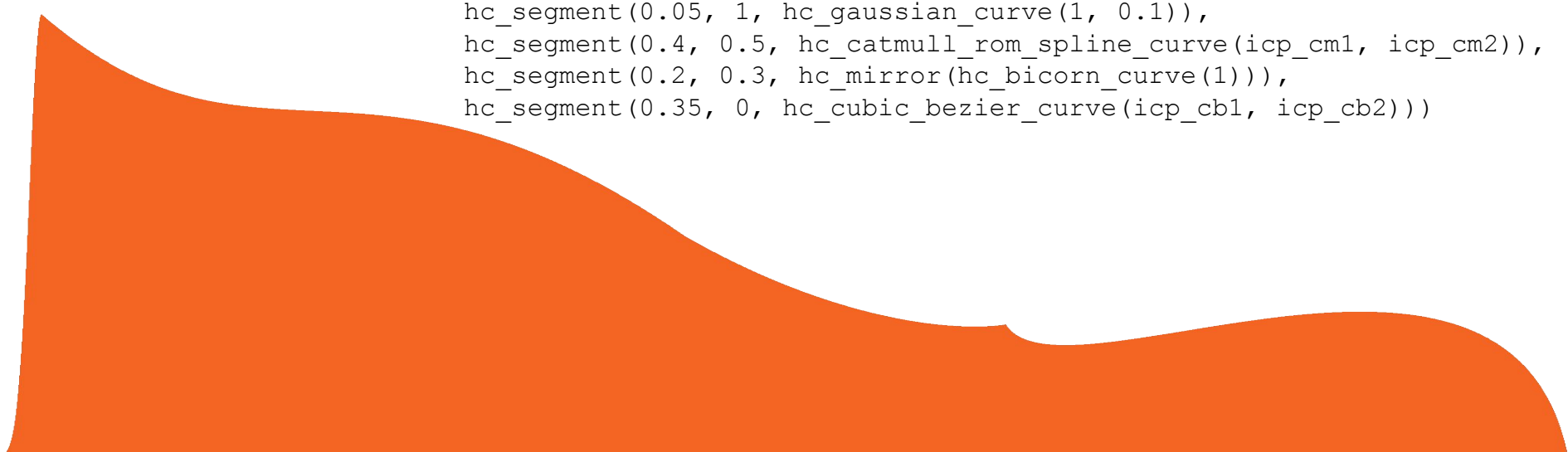
gi5 → Simple AR envelop

```
gienvdur        init 8192

; hc_hypercurve(function_nbr, number_of_samples, y_start_point, segment1,
[segment2…])
; hc_segment(fractional_size, y_destination, curve_base)

gi5             hc_hypercurve 0, gienvdur, 0,
                     hc_segment(0.1, 1, hc_diocles_curve(0.5001)),
                     hc_segment(0.9, 0, hc_diocles_curve(0.55))
```



gi6 → Complex curve

```
gienvdur       init 8192

; Control points for catmull rom curve
icp_cm1 = hc_control_point(-1, 3)
icp_cm2 = hc_control_point(3, -2)
; Control points for cubic bezier curve
icp_cb1 = hc_control_point(0.1, 0.1)
icp_cb2 = hc_control_point(0.9, 0.65)

gi6            hc_hypercurve 0, gienvdur, 0,
                    hc_segment(0.05, 1, hc_gaussian_curve(1, 0.1)),
                    hc_segment(0.4, 0.5, hc_catmull_rom_spline_curve(icp_cm1, icp_cm2)),
                    hc_segment(0.2, 0.3, hc_mirror(hc_bicorn_curve(1))),
                    hc_segment(0.35, 0, hc_cubic_bezier_curve(icp_cb1, icp_cb2)))
```



Future of HYPERCURVE

- Make music with it
- Experiment
- Implement new curves, and use them in new ways
- New manipulation tools : segment subdivision, curve interpolation, curve extraction, curve against curve symmetry, virtual 3D axis
- Thoughts on downsampling

jacopo.grecodalceo@gmail.com

johannphilippe@lilo.org

~ thank you ~

github.com/johannphilippe/hypercurve