Curvature combs and harmonized paths in METAPOST

Linus Romer

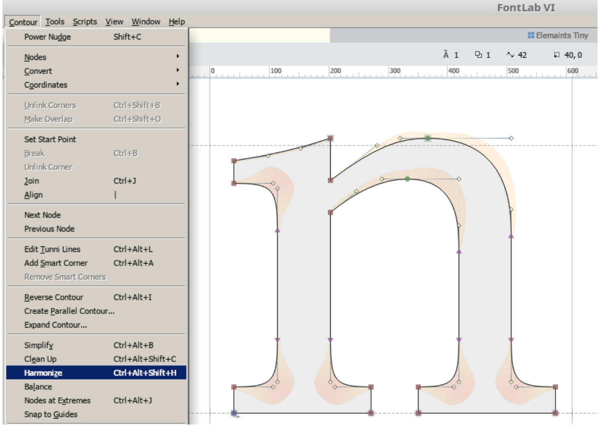
July 15, 2023

Introduction

Popular in font design software:

- curvature combs
- harmonization

⇒ Implementation in METAPOST



Introduction

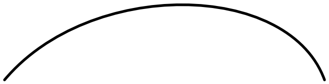
How I got into it:

- I designed typefaces with METAFONT (e.g. *Funtauna*) and in FontForge (e.g. *Miama Nueva*)
- ⇒ conversion tool `mf2outline`
- ⇒ some tool implementations in FontForge (e.g. *Harmonize*, *Balance*)

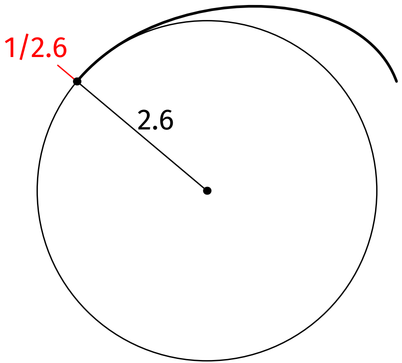
Curvature

$$\text{curvature} = \frac{1}{\text{radius of osculating circle}}$$

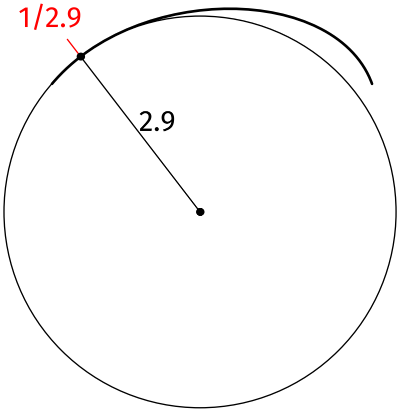
Curvature along a curve



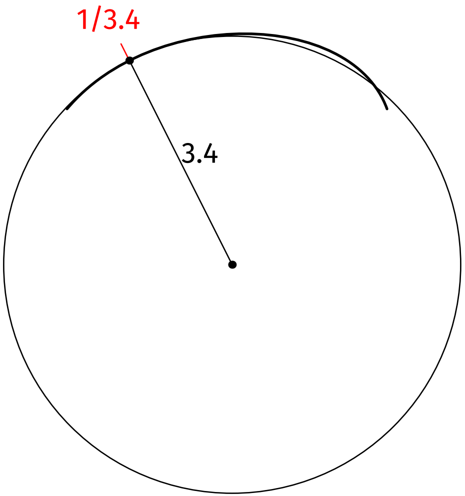
Curvature along a curve



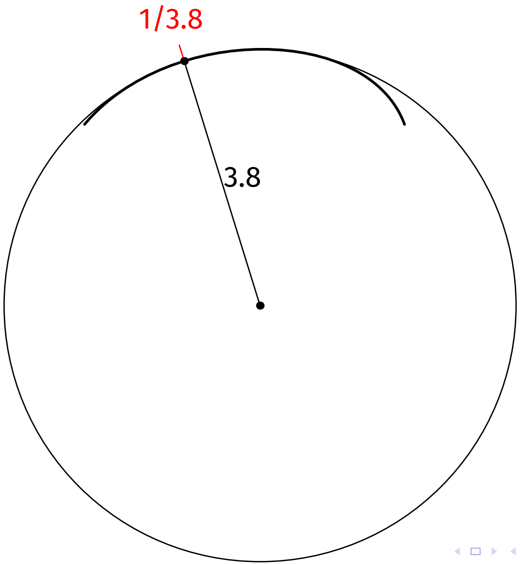
Curvature along a curve



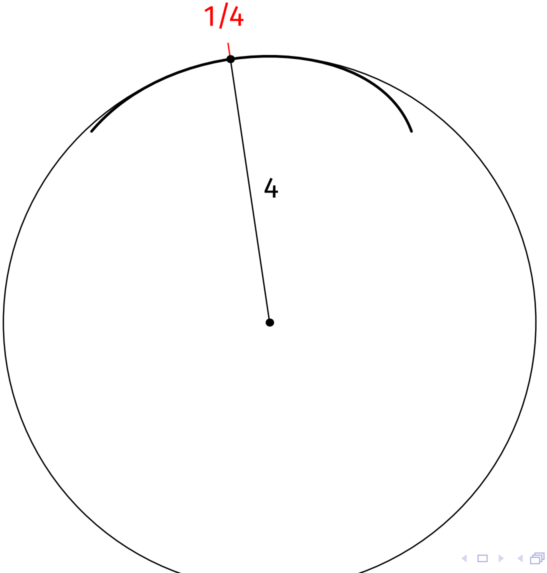
Curvature along a curve



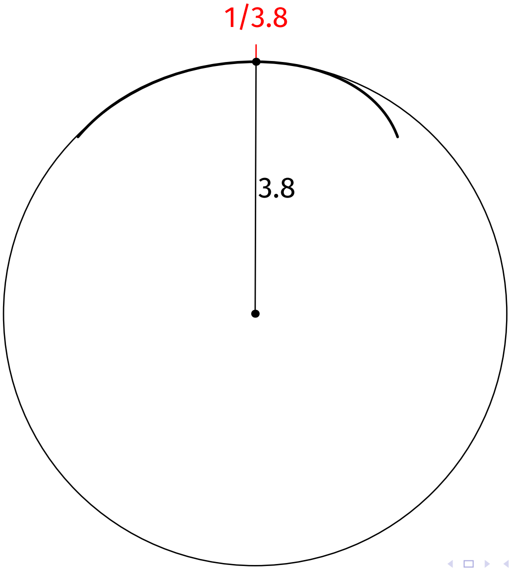
Curvature along a curve



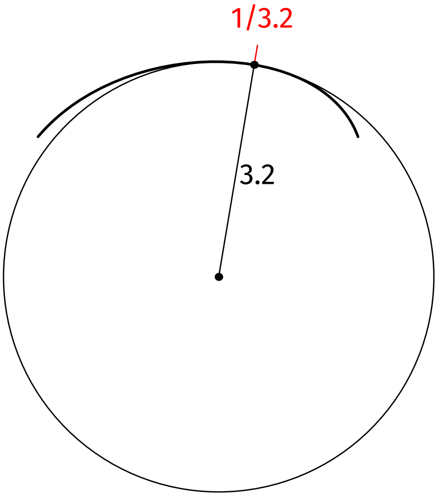
Curvature along a curve



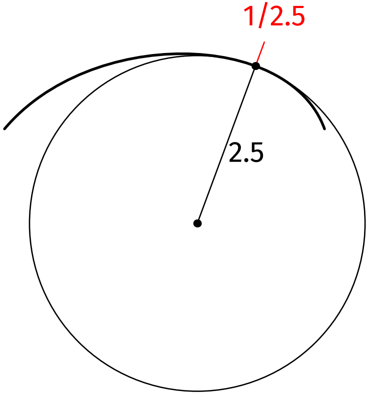
Curvature along a curve



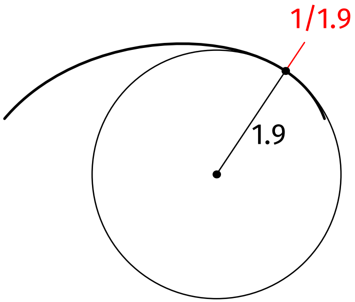
Curvature along a curve



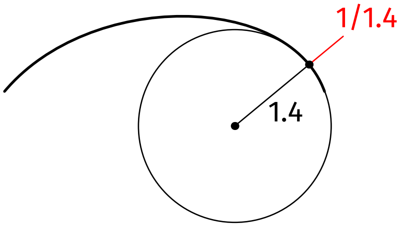
Curvature along a curve



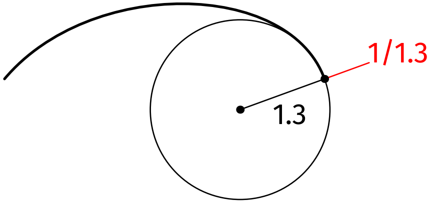
Curvature along a curve



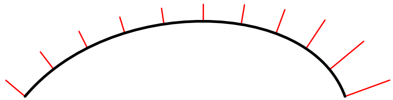
Curvature along a curve



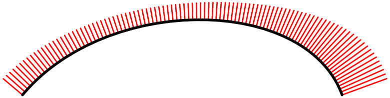
Curvature along a curve



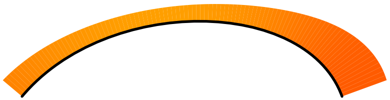
Curvature comb



Curvature comb

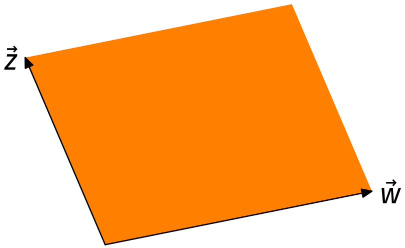


Curvature comb



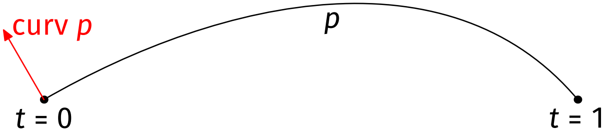
Cross product macro

```
primarydef w crossprod z =
 (xpart w * ypart z - ypart w * xpart z)
enddef;
```



Macro for initial curvature

```
vardef curv expr p =
```



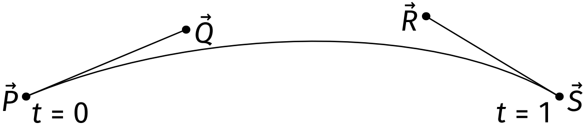

```
enddef;
```

- `curv t of p` would be a bad idea for integer t's!

Math of curvature

Cubic Bézier curve:

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = t^3(3\vec{Q} - \vec{P} + \vec{S} - 3\vec{R}) + 3t^2(\vec{P} - 2\vec{Q} + \vec{R}) + 3t(\vec{Q} - \vec{P}) + \vec{P}$$



The initial derivatives are then:

$$\begin{pmatrix} \dot{x}(0) \\ \dot{y}(0) \end{pmatrix} = 3\underbrace{(\vec{Q} - \vec{P})}_{=:\vec{v}} \qquad \begin{pmatrix} \ddot{x}(0) \\ \ddot{y}(0) \end{pmatrix} = 6\underbrace{(\vec{P} - 2\vec{Q} + \vec{R})}_{=:\vec{w}}$$

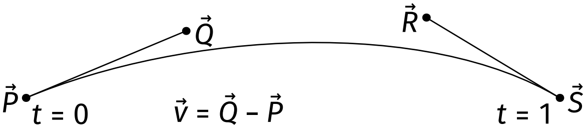
Math of curvature

$$\text{initial curvature} = \frac{\begin{pmatrix}\dot{x}(0)\\ \dot{y}(0)\end{pmatrix} \times \begin{pmatrix}\ddot{x}(0)\\ \ddot{y}(0)\end{pmatrix}}{\left|\begin{pmatrix}\dot{x}(0)\\ \dot{y}(0)\end{pmatrix}\right|^3} = \frac{3\vec{v} \times 6\vec{w}}{(3l)^3}$$

$$= \frac{2}{3}\frac{\vec{v} \times \vec{w}}{l^3} = \frac{2}{3l} \cdot \left(\frac{1}{l}\vec{v} \times \frac{1}{l}\vec{w}\right)$$

- divisions by l: prevent arithmetic overflow
- special case $\left|\begin{pmatrix}\dot{x}(0)\\ \dot{y}(0)\end{pmatrix}\right| = 0$ not handled here: $\Rightarrow$ curvature probably $\pm\infty$

Macro for the initial curvature



$$\vec{v} = \vec{Q} - \vec{P}$$

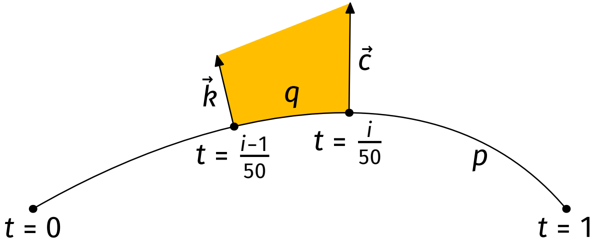
$$\vec{w} = \vec{P} - 2\vec{Q} + \vec{R}$$

$$\text{initial curvature} = \frac{2}{3l} \cdot \left(\frac{1}{l}\vec{v} \times \frac{1}{l}\vec{w}\right) \qquad \text{with } l = |\vec{v}|$$

```
vardef curv expr p =
 save v,w,l; pair v,w;
 v = direction 0 of p;
 l = length v; v := v/l;
 w = (point 0 of p - 2*postcontrol 0 of p
                    + precontrol 1 of p)/l;
 2/3*(v crossprod w)/l*(v rotated -90)
enddef;
```

Macro for the curvature comb

- subdivide each segment into 50 subpaths q
- Each part of the comb is made of two subsequent "curvature" vectors $\vec{k}$, $\vec{c}$ (scaled by a constant factor s)
- color depends on the average length of $\vec{k}$ and $\vec{c}$



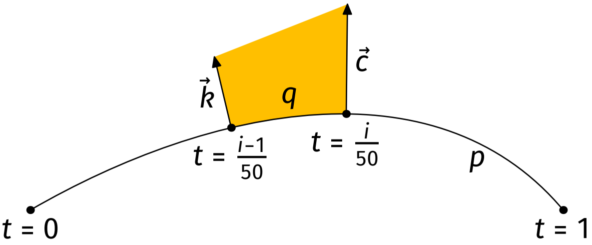
Macro for the curvature comb

```
vardef comb(expr p,s) =
 save q,c,k; path q; pair c,k;
 for n = 0 upto length(p)-1:
  c := s * curv subpath(n,n+1) of p;
  for i = 1 upto 50:
   k := c;
   c := s * curv subpath(n+i/50,n if i<25: +1 fi) of p;
   q := subpath(n+(i-1)/50,n+i/50) of p;
   fill q -- point 1 of q + c
          -- point 0 of q + k
          -- cycle withcolor
    (1,1/(1+.1*.5[length c,length k]),0);
  endfor
 endfor
enddef;
```

Macro for the curvature comb

The condition `if i<25: +1 fi` makes the subpath for the calculation of the curvature as large as possible to be more accurate:

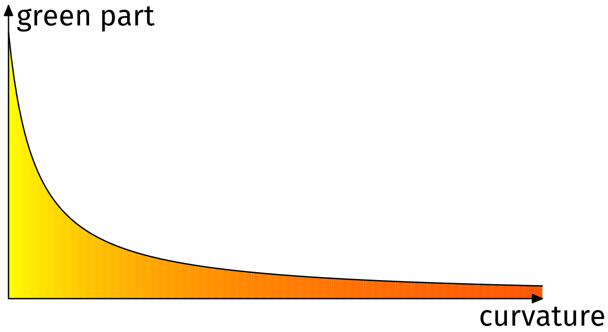
```
c := s * curv subpath(n+i/50,
         n if i<25: +1 fi) of p;
```



Macro for the curvature comb

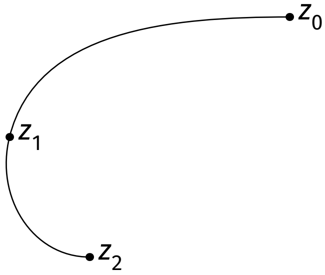
- curvature 0 $\mapsto$ yellow
- curvature $\pm\infty$ $\mapsto$ red
- done by changing the green value between 1 and 0
- increase the `.1` to make curvature more red

```
(1,1/(1+.1*.5[length c,length k]),0)
```



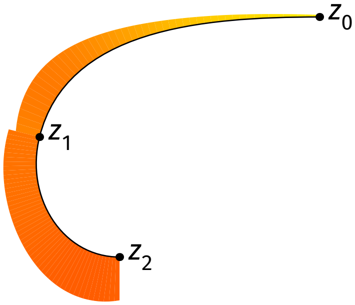
Harmonize paths

```
z0{left} .. z1 .. z2{right}
```



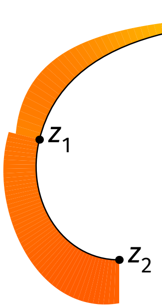
Harmonize paths

```
z0{left} .. z1 .. z2{right}
```



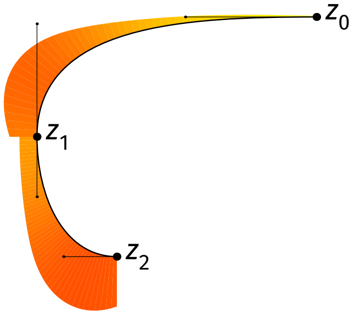
Harmonize paths

direction in z_1 was chosen to have continuous *mock curvature* [Hobby, 1986]



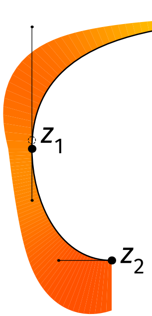
Harmonize paths

```
z0{left} .. z1{down} .. z2{right}
```



Harmonize paths

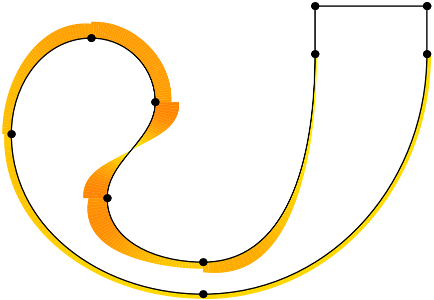
```
harmonize z0{left} .. z1{down} .. z2{right}
```



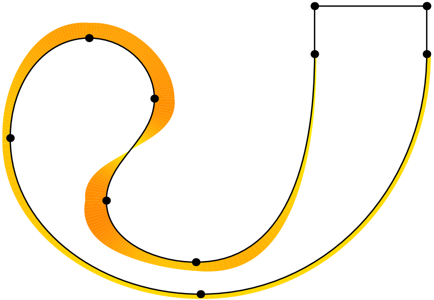
Examples of harmonization

Harmonized paths do not necessarily look better...

...but they might!



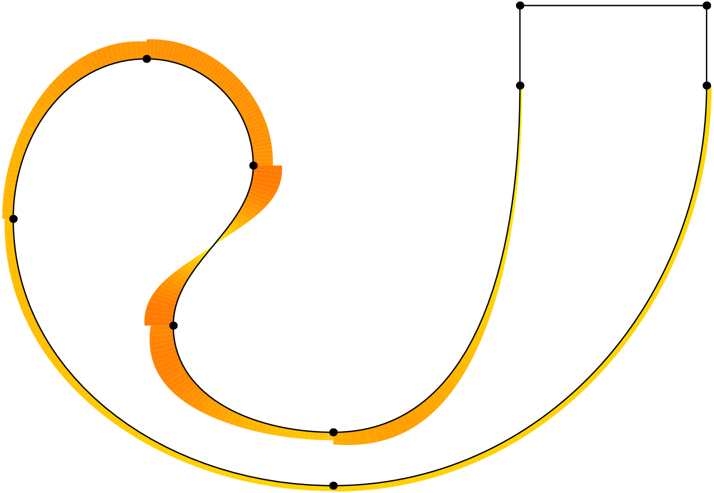
not harmonized



harmonized

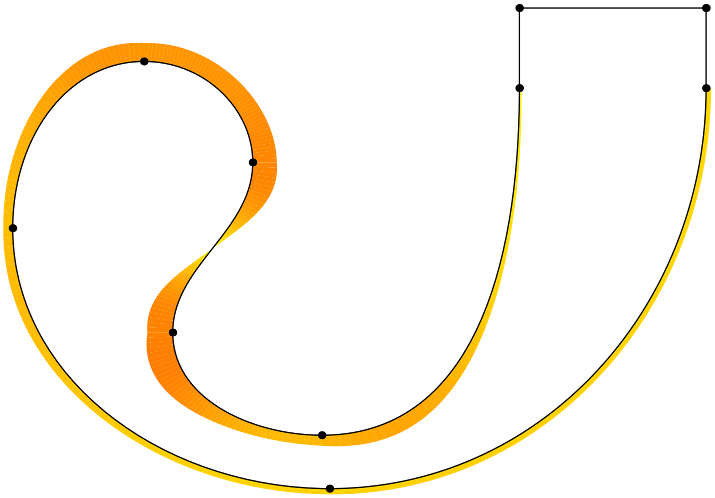
Examples of harmonization

```
   (130,90)
-- (130,75){down}
.. (60,0){left}
.. (0,50){up}
.. (25,80){right}
.. (45,60){down}
.. (30,30){down}
.. (60,10){right}
.. (95,75){up}
-- (95,90)
-- cycle
```

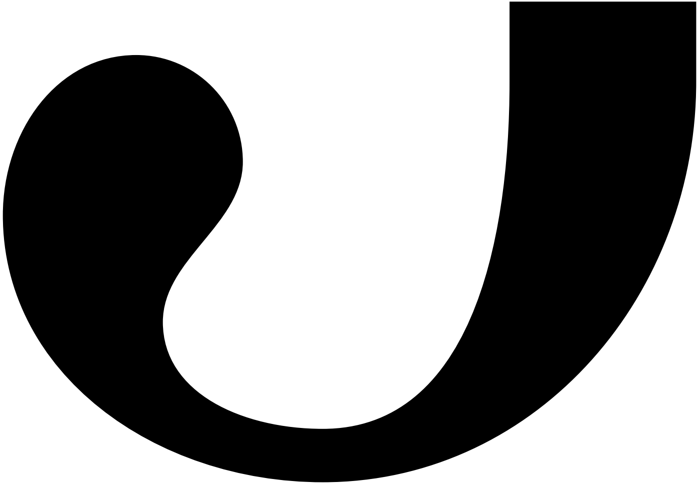


not harmonized

Examples of harmonization

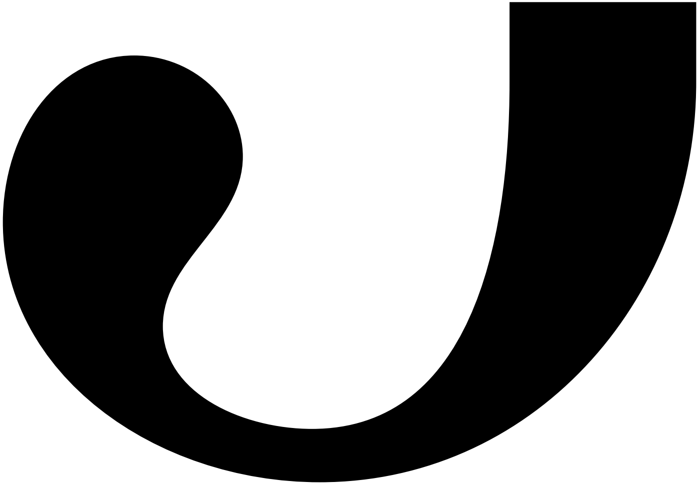


Examples of harmonization



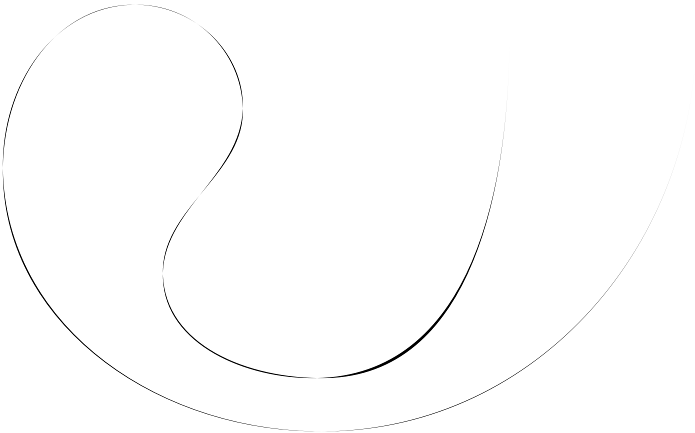
not harmonized

Examples of harmonization



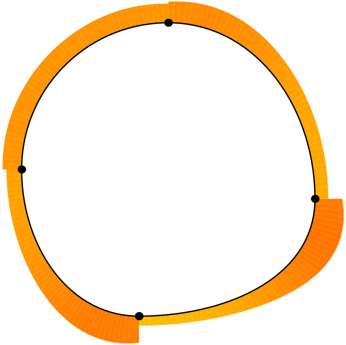
harmonized

Examples of harmonization

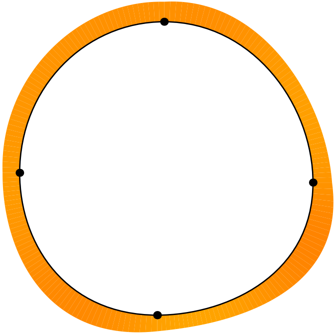


Examples of harmonization

Harmonized paths tend to be more rounded.



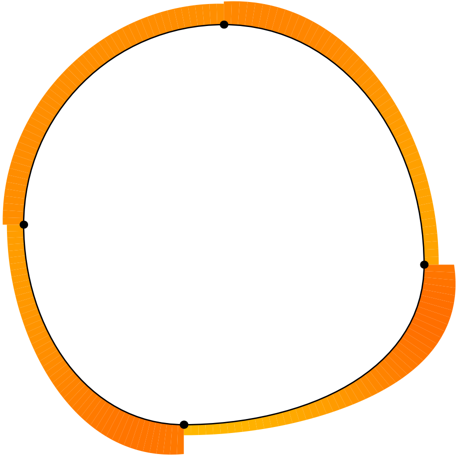
not harmonized



harmonized

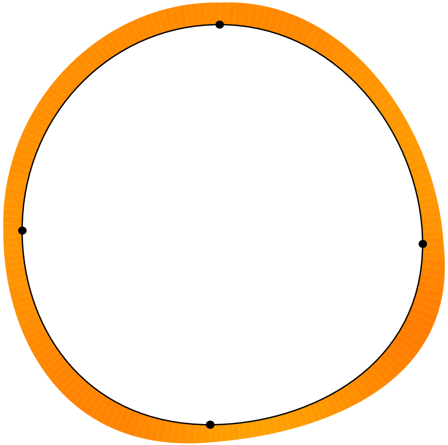
Examples of harmonization

```
   (0,0){right}
.. (30,20){up}
.. (5,50){left}
.. (-20,25){down}
.. cycle
```



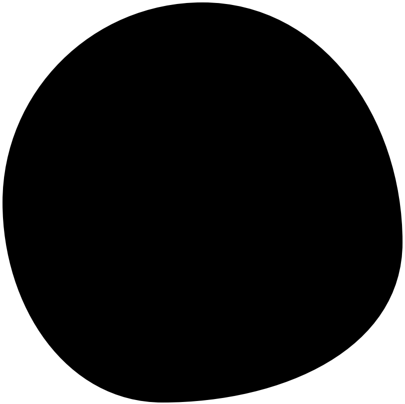
not harmonized

Examples of harmonization



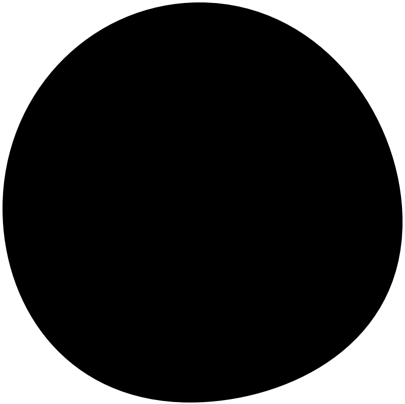
harmonized

Examples of harmonization



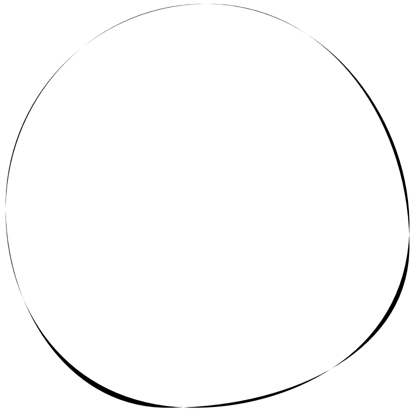
not harmonized

Examples of harmonization



harmonized

Examples of harmonization

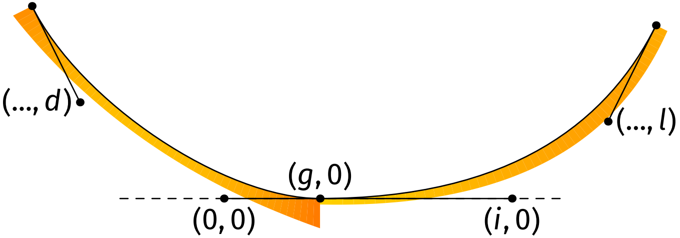


Math of harmonization (generic case)

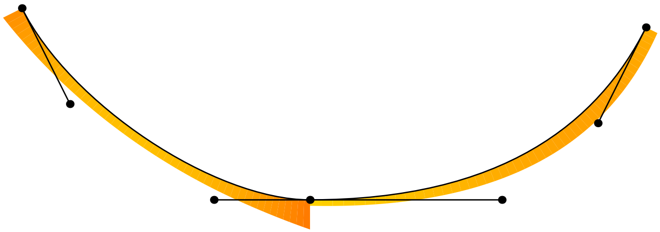
Assume two adjoint cubic Bézier curves:

- same direction in their joint
- no zero-handles
- no inflection point in their joint

Whitout loss of generalization:

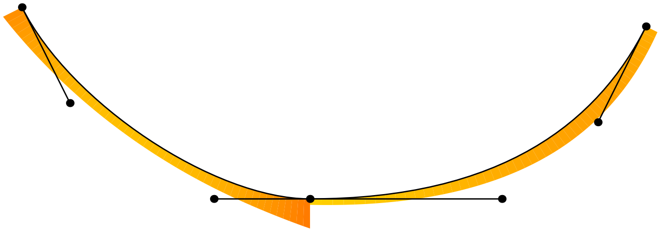


Math of harmonization (generic case)

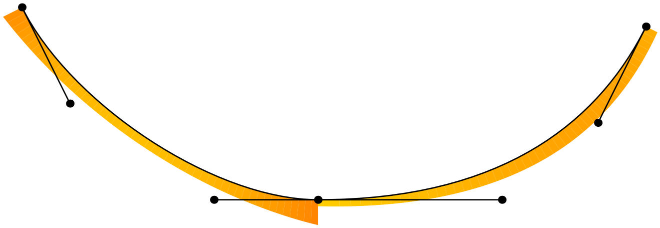


Goal: Move the joining node between its control points sucht that the curvature becomes continuous.

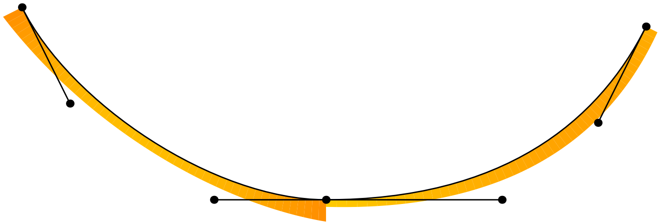
Math of harmonization (generic case)



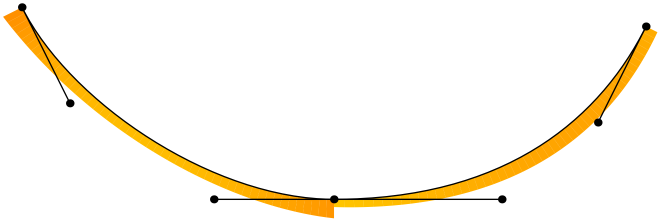
Math of harmonization (generic case)



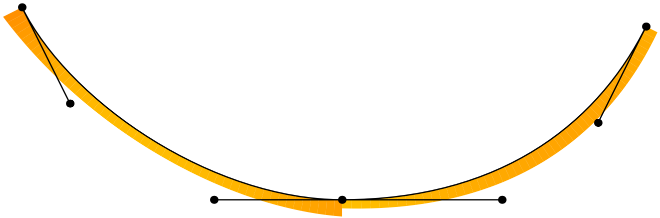
Math of harmonization (generic case)



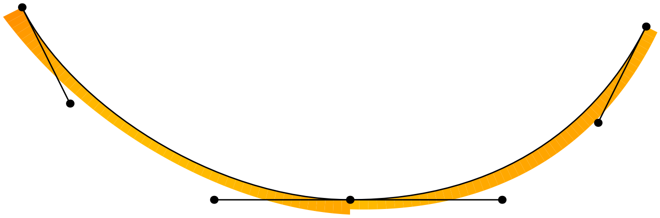
Math of harmonization (generic case)



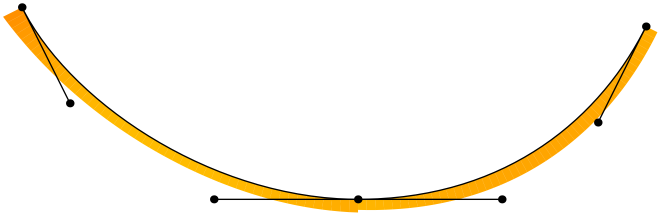
Math of harmonization (generic case)



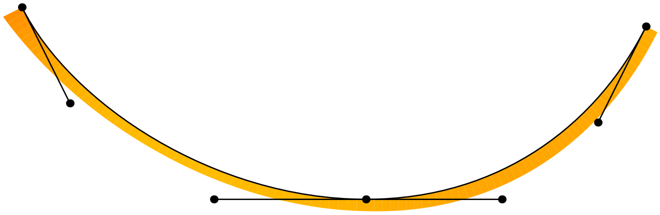
Math of harmonization (generic case)



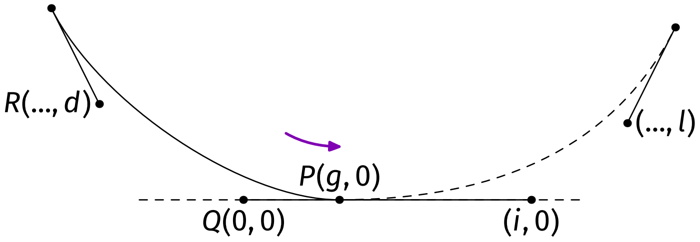
Math of harmonization (generic case)



Math of harmonization (generic case)



Math of harmonization (generic case)

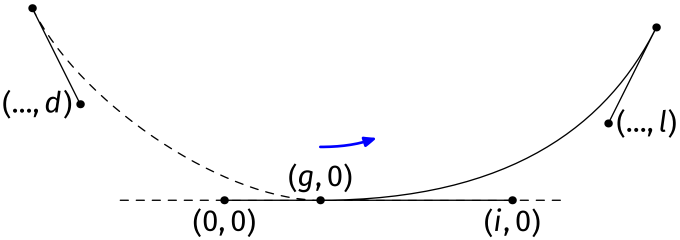


$$\vec{v} = \vec{Q} - \vec{P} = \begin{pmatrix} -g \\ 0 \end{pmatrix} \qquad \vec{w} = \vec{P} - 2\vec{Q} + \vec{R} = \begin{pmatrix} g + \ldots \\ d \end{pmatrix}$$

curvature in P to the left heading right $= -\dfrac{2}{3|\vec{v}|^3} \cdot \underbrace{\vec{v} \times \vec{w}}_{-gd}$

$$\overset{g>0}{=} \frac{2d}{3g^2}$$

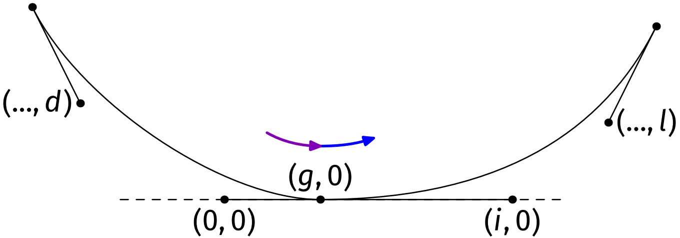
Math of harmonization (generic case)



Analogous:

$$\text{curvature in } P \text{ to the right heading right} \overset{i>g}{=} \frac{2l}{3(i-g)^2}$$

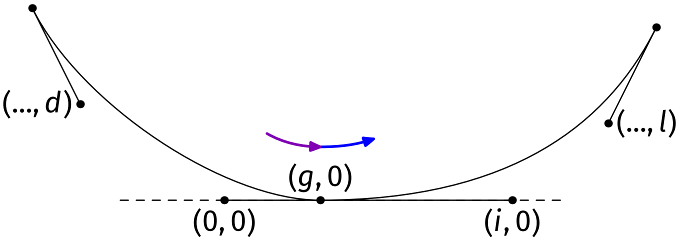
Math of harmonization (generic case)



Equal curvatures in $(g, 0)$:

$$\frac{2d}{3g^2} = \frac{2l}{3(i-g)^2} \quad \Leftrightarrow \quad g = \begin{cases} \dfrac{d \pm \sqrt{dl}}{d-l} \cdot i & \text{if } d \neq l, \\ \dfrac{i}{2} & \text{else.} \end{cases}$$

Math of harmonization (generic case)

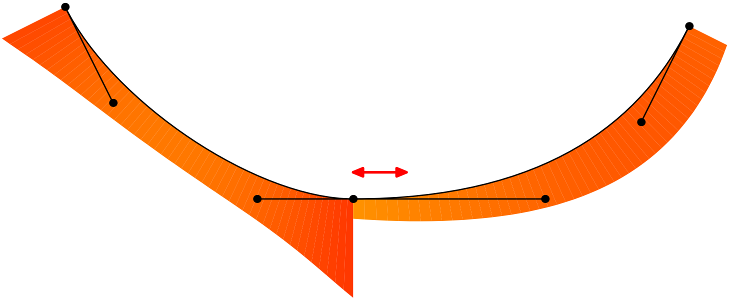


$$g = \begin{cases} \dfrac{d \pm \sqrt{dl}}{d - l} \cdot i & \text{if } d \neq l, \\ \dfrac{i}{2} & \text{else.} \end{cases}$$

Geometric mean $\sqrt{dl}$ $\Rightarrow$ new joining knot is between its control points

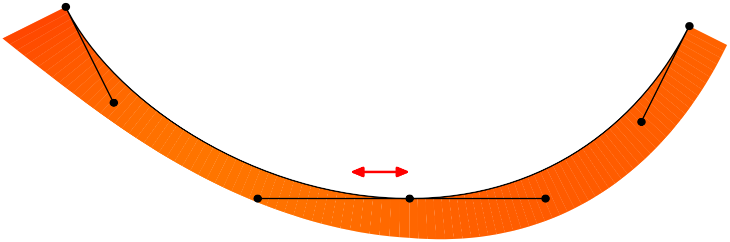
Math of harmonization (generic case)

- The curvatures at the other ends will not change!
- $\Rightarrow$ global solution = sum of local solutions



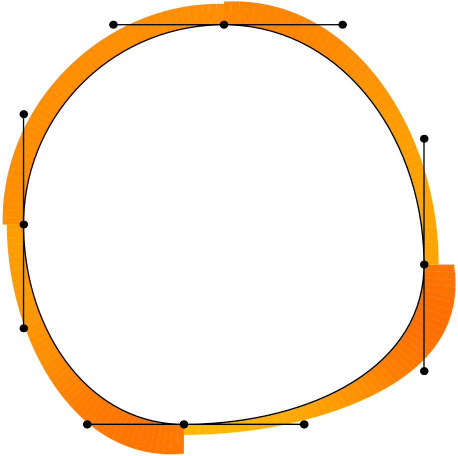
Math of harmonization (generic case)

- The curvatures at the other ends will not change!
- $\Rightarrow$ global solution = sum of local solutions



Math of harmonization (generic case)

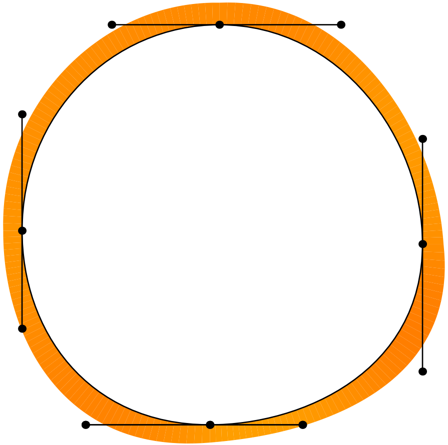
Example of a global solution for a continuous curvature:



not harmonized

Math of harmonization (generic case)

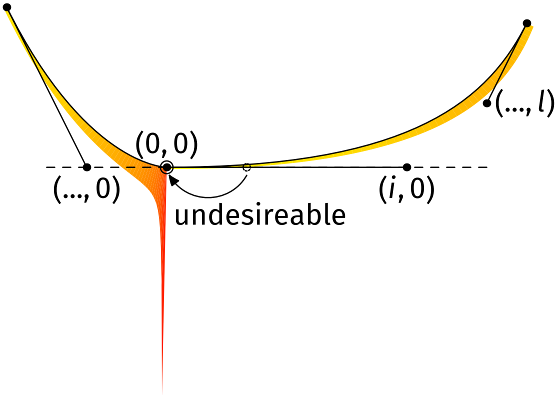
Example of a global solution for a continuous curvature:



harmonized

Math of harmonization (special case I)

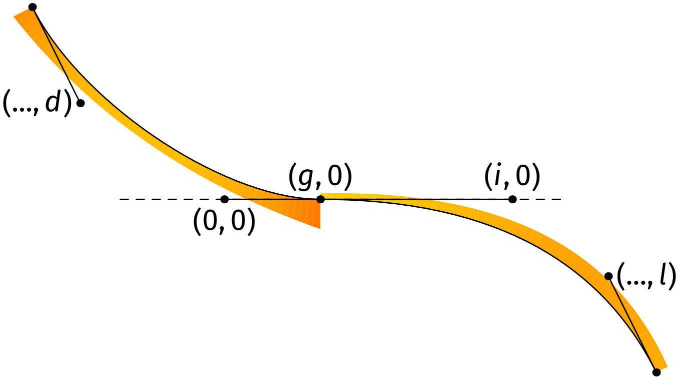
- If either d or l is zero, $g = \frac{d-\sqrt{dl}}{d-l} \cdot i$ becomes either 0 or i
- $\Rightarrow$ joining knot will become collocated with one of its control points
- $\Rightarrow$ curvature might become infinitely large!



Math of harmonization (special case II)

Assume two adjoint cubic Bézier curves:

- same direction in their joint
- no zero-handles
- inflection point in their joint

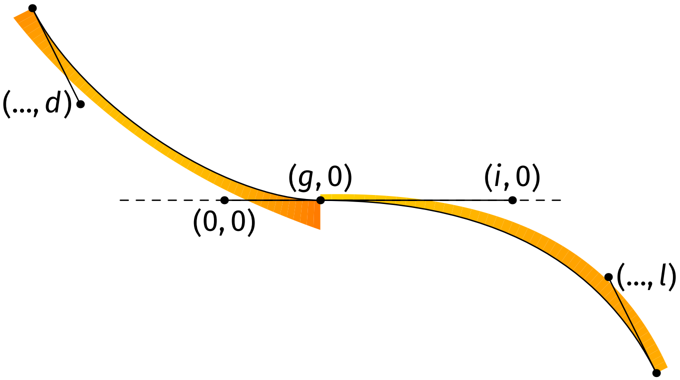


Math of harmonization (special case II)

Inflection point

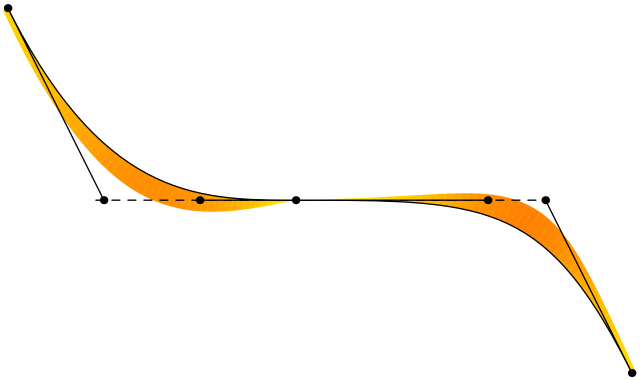
$\Rightarrow$ curvatures $\dfrac{2d}{3g^2}$ and $\dfrac{2l}{3(i-g)^2}$ must have different signs!

$\Rightarrow$ $d = l = 0$ for curvature-continuous solutions



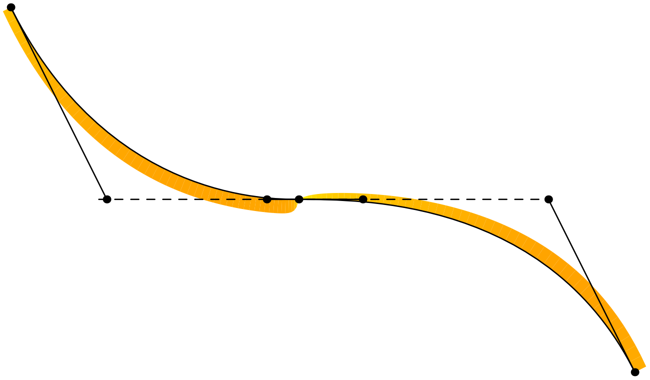
Math of harmonization (special case II)

$\Rightarrow$ All control points must lie on one line for curvature-continuous solutions:



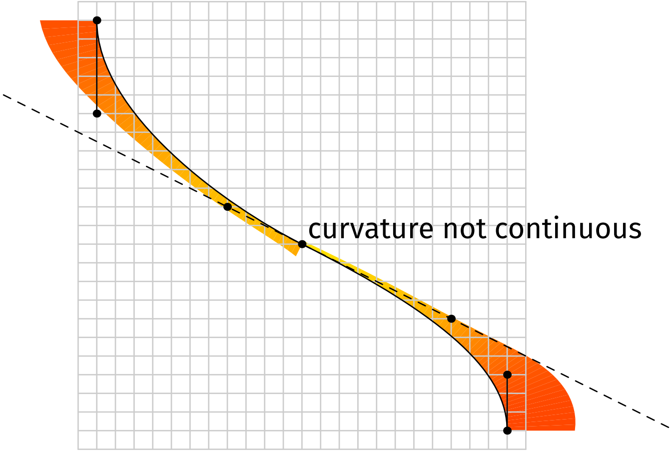
Math of harmonization (special case II)

Further conditions (like preservation of area) could be satisfied by additionally moving the control points of the joining knot:



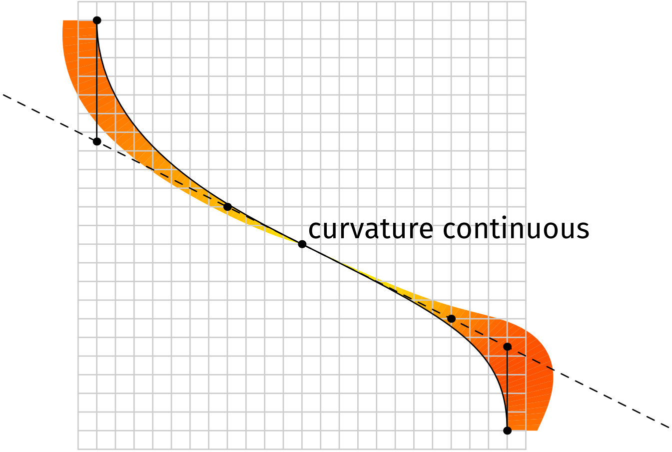
Math of harmonization (special case II)

Rounding problem on grid:



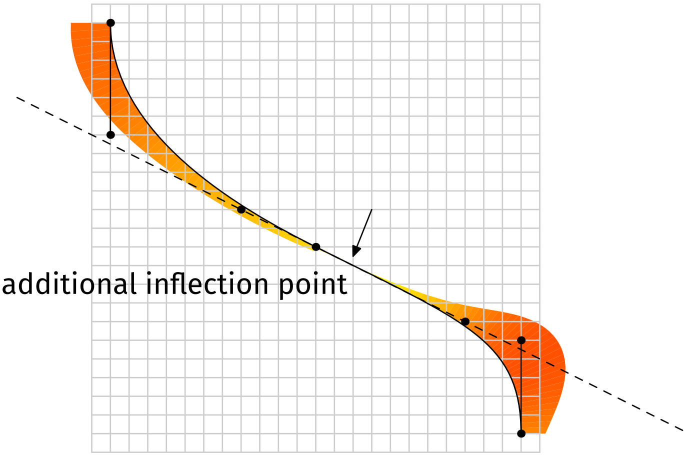
Math of harmonization (special case II)

Rounding problem on grid:



Math of harmonization (special case II)

Rounding problem on grid:



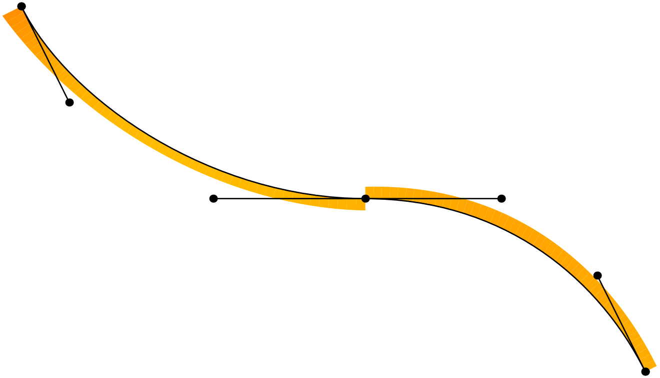
Math of harmonization (special case II)

Problems of a curvature continuous solution for a joining knot with an inflection:

- rounding errors may introduce new inflection points (wobbly curve)
- off-curve points are changed ⇒ different behaviour than before

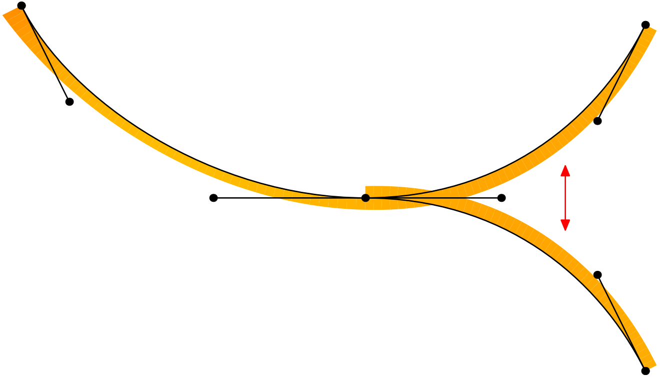
Math of harmonization (special case II)

These problems are solved, if we only guarantee the *absolute value* of the curvature to be continuous in this case.



Math of harmonization (special case II)

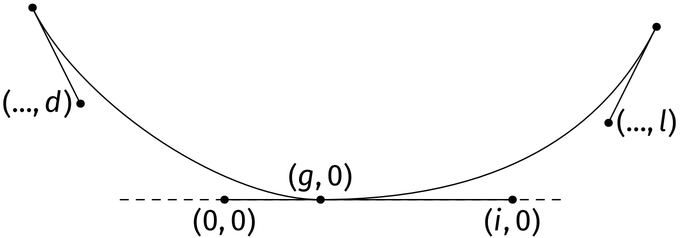
This is just the former solution mirrored.



Math of harmonization

Definition: *harmonization* is the act of setting

$$g_{\text{new}} = \begin{cases} g_{\text{old}} & \text{if } d = 0 \text{ or } l = 0, \\ \frac{i}{2} & \text{else if } |d| = |l|, \\ \frac{|d| - \sqrt{|dl|}}{|d| - |l|} \cdot i & \text{else} \end{cases}$$



Harmonization macro

- Iterating through the joining knots of a path p:
  - check if the joint is smooth
  - calculate new position g_{new} between control points and store it as a new point in an array
- If cyclic: Last point = first new point
- return a path made of the new joining knots and old control points

Harmonization macro

```
vardef harmonize expr p =
 save t,u,d,l,n,q; pair t,u,q[];
 n = length p;
 for j = if cycle p: 0 else: 1 fi upto n-1:
  q[j] = point j of p;
  t := unitvector(direction j of p);
  u := unitvector(point j of p - precontrol j of p);
  if eps > abs((u dotprod t) - 1):
...
```

Harmonization macro

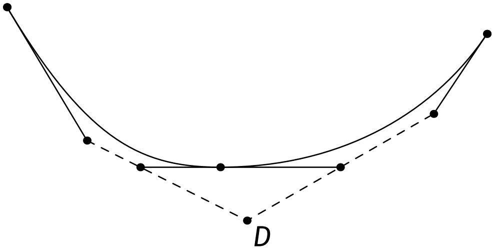
```
...
    l := abs(t crossprod (precontrol j+1 of p
                              - point j of p) );
    d := abs(t crossprod (postcontrol j-1 of p
                               - point j of p) );
    if not ( (l = 0) or (d = 0) ):
     q[j] :=
      if (d = l):
       .5
      else:
       ((d-sqrt(d*l))/(d-l))
      fi
      [precontrol j of p,postcontrol j of p];
    fi
   fi
  endfor
```

Harmonization macro

```
...
 if not cycle p:
  q[0] = point 0 of p;
  q[n] = point n of p;
 fi
 q[0]
 for j = 0 upto n-1:
  .. controls postcontrol j of p
  and precontrol j+1 of p .. if (j = n-1)
  and (cycle p): cycle else: q[j+1] fi
 endfor
enddef;
```

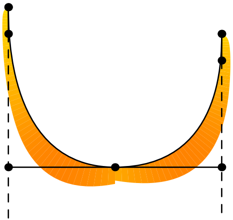
History of Harmonization

- 1990: Robert L. Roach and John R. Forrest publish an algorithm to reach curvature continuity for cubic bézier curves [Roach, 1990]
- the algorithm is equivalent to the presented algorithm, but depends on an intersection point D, which may not exist



History of Harmonization

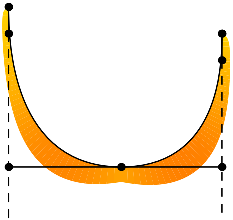
- 1990: Robert L. Roach and John R. Forrest publish an algorithm to reach curvature continuity for cubic bézier curves [Roach, 1990]
- the algorithm is equivalent to the presented algorithm, but depends on an intersection point *D*, which may not exist



not harmonized

History of Harmonization

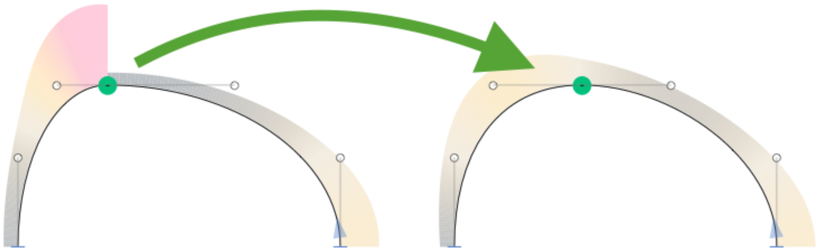
- 1990: Robert L. Roach and John R. Forrest publish an algorithm to reach curvature continuity for cubic bézier curves [Roach, 1990]
- the Roach-algorithm is equivalent to the presented algorithm, but depends on an intersection point D, which may not exist



harmonized

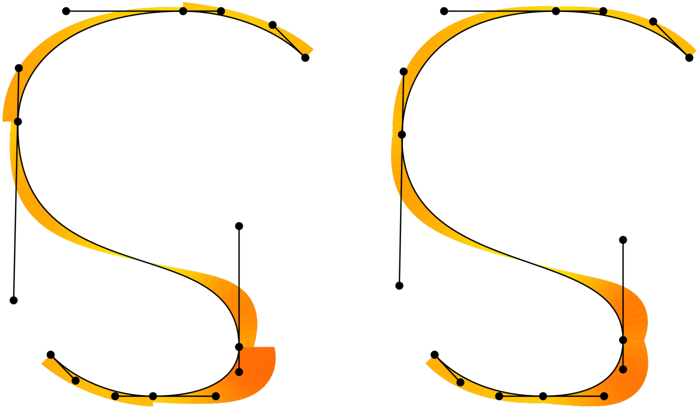
History of Harmonization

- The *supertool* plugin for the Glyphs app uses a slightly modified Roach-algorithm
- The Green harmony plugin for Glyphs app uses the Roach-algorithm



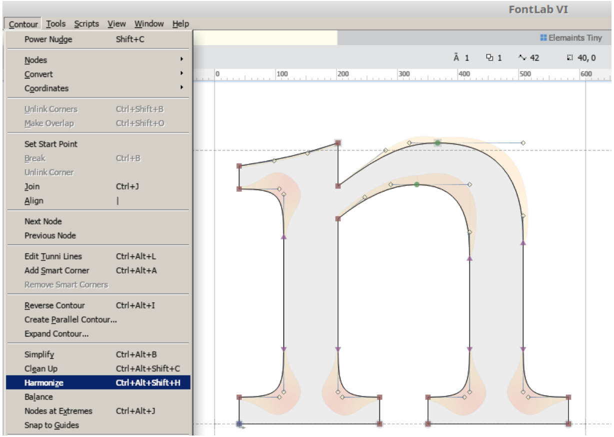
History of Harmonization

- 2010: The *Font Remix Tools 1.6* (plugin for Fontlab and Glyphs) by *Just Another Foundry* introduces a «Harmonizer» (closed source)
- The here defined *harmonization* algorithm seems to be different



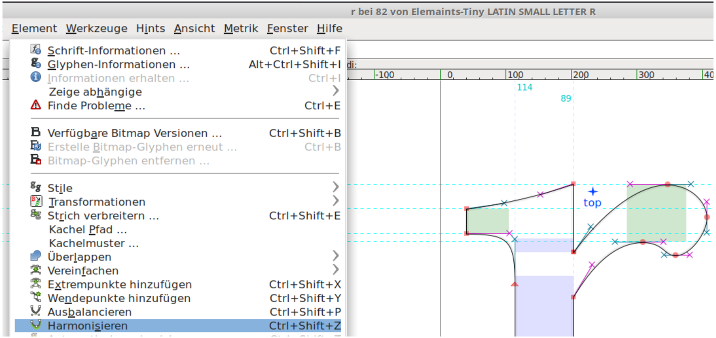
History of Harmonization

- 2017: Fontlab VI introduces «harmonize» (closed source)
- The here defined *harmonization* algorithm may differ (acts similar)



History of Harmonization

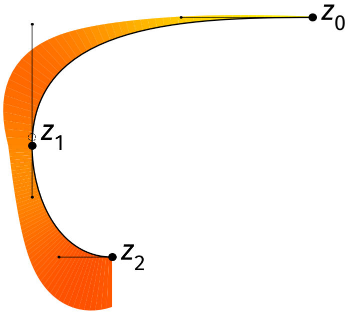
- 2019: plug-ins *harmonize-tunnify-inflection* and *curvatura* for FontForge implement the presented harmonization algorithm
- 2022: the presented harmonization algorithm becomes part of FontForge itself



Smoothing out paths even more

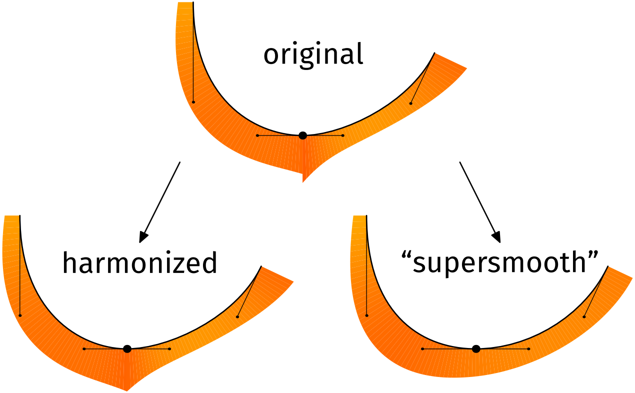
Disadvantage of harmonization:

Harmonized paths normally no longer interpolate the points they were originally meant to!



Smoothing out paths even more

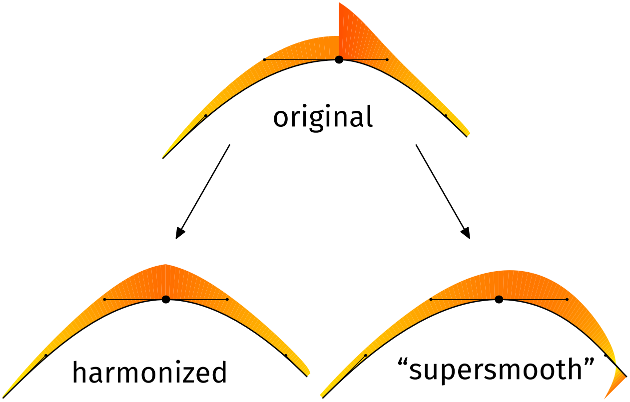
Moving the control points open more possibilites: Also the change of curvature can be made continuous («supersmooth»)



Smoothing out paths even more

Bad idea:

- supersmoothness may introduce new inflection points
- hard to globalize for more than two segments



References I

Hobby, J. D. (1986).
Smooth, easy to compute interpolating splines.
Discrete & computational geometry, 1(2):123–140.

Roach, R. L. (1990).
Curvature continuity of cubic bezier curves in the solid modeling aerospace research tools design software.
interim report, NASA Langley Research Center.