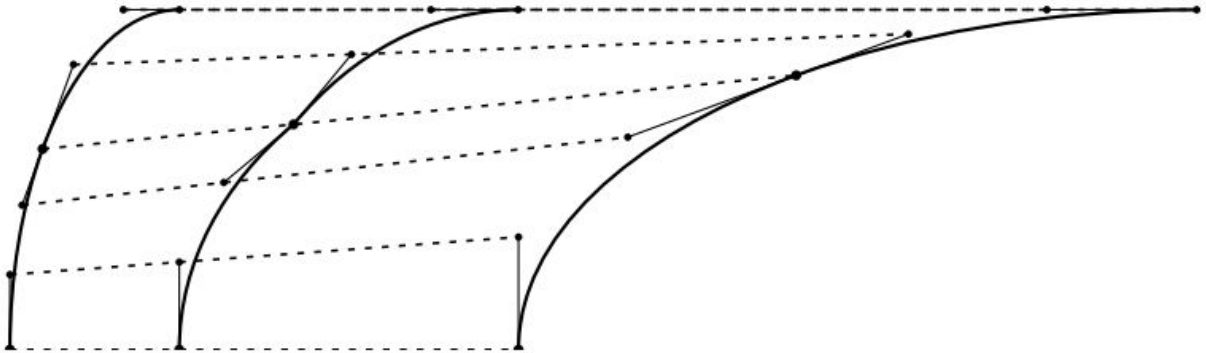
A new spline

Raph Levien and Jacob Rus

Libre Graphics Meetup • Saarbrücken 2019

Why a new spline?

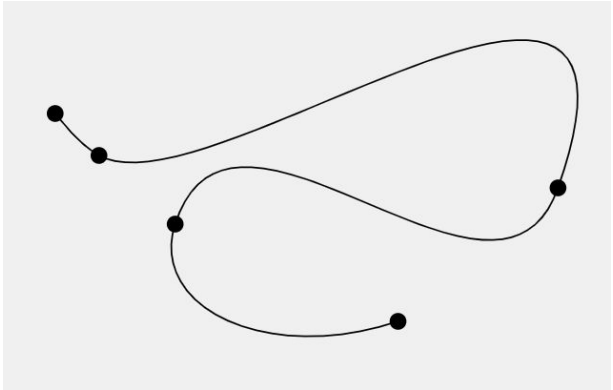
- Béziers are expressive but not “natural”
  - Difficult to learn
  - When interpolating, can form kinks
    - Not G2-continuous by construction



- Thus, many (interpolating) splines in the literature

Hobby Spline (1986)

- Originally designed for Metafont
- Also used in Apple iWork
- Is approximately G2-continuous, so interpolates well
- Numerically robust, but tangents flip direction
- Ugly shapes at large angles of control polygon



- Online demo: weitz.de/hobby/

Spiro spline (2007)

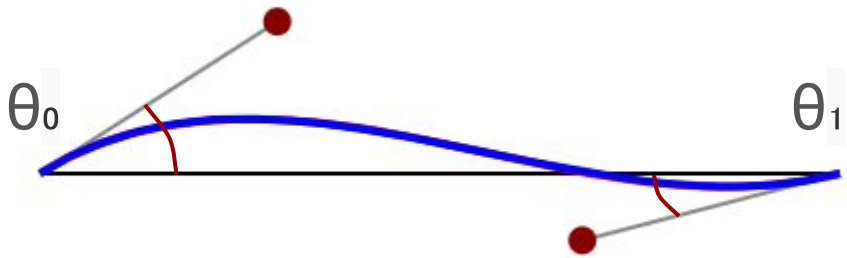
- Based on Euler spiral spline (KURGLA 2, 1974)
  - The spiral itself dates to 1744
- Inconsolata designed using this spline
- Implemented in FontForge, Inkscape
- Very smooth curvature
- Not numerically robust, can go wild
- Difficult to make good UX
- Online demo: levien.com/garden/js/spiro.html

Adobe κ-Curves (2017)

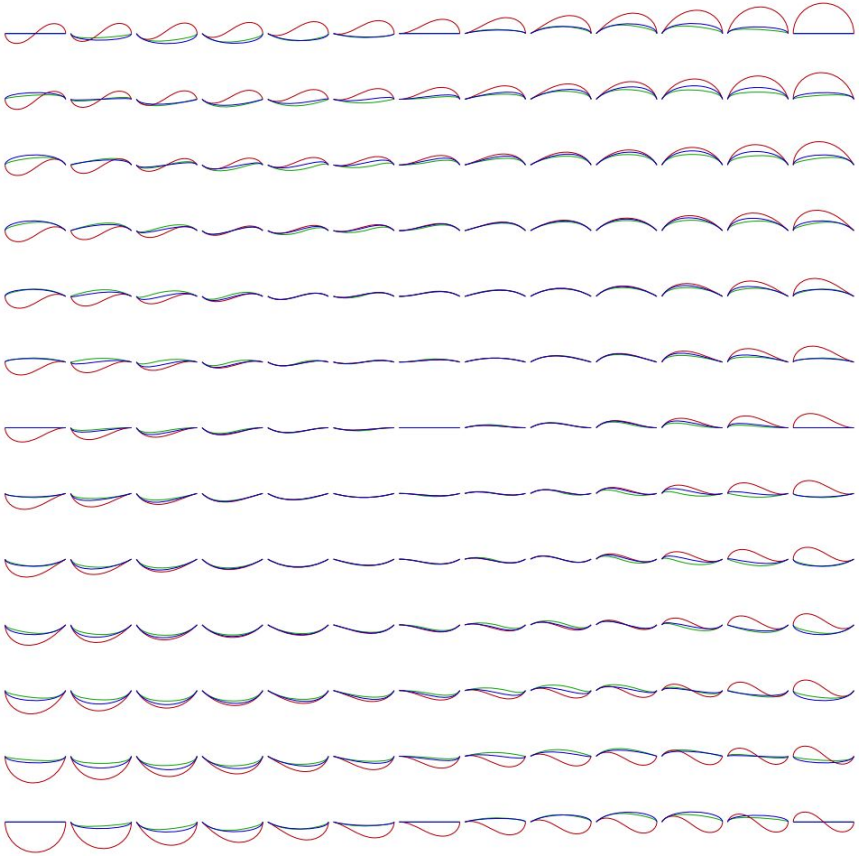
- Solves numerical robustness problem
  - Always a solution, always unique
- Solves “flipping” – small changes to input → small changes to result
- Flip side: it sometimes forms “cusps”
  - A good tradeoff for interactive use
- Version in paper uses parabola (quadratic Bézier) segments
  - No true inflection (curvature discontinuity in S shapes)
  - Lumpy curvature, not close to a circle
  - Note: version shipping in Adobe tools is different
    - differences are not documented publicly

2 parameter splines

- Derive spline from a 2-parameter family of curves
- The two parameters are angle relative to chord at both endpoints
- Spline is a global solution of tangents so curvature is continuous
- Only depends on curvature at endpoints, not shape of interior



Same framework, different curves



Existing 2 parameter splines

- Minimum Energy Curve (idealized flexible strip spline)
- Piecewise Scale-Invariant MEC (KURGLA 1, Mehlum 1969)
- Euler spiral spline
- Log-aesthetic spiral spline (CAD 2009)
- Séquin & Lee circle spline (2005)
- κ-Curves
- Hobby spline (approximate)
    - Curve family is 2-parameter, but curvature uses linear approximation

Many different properties

- Minimum energy: MEC
- Roundness: SI-MEC, Euler spiral, circle spline
- Locality: circle spline
- Unique solution: κ-Curves, Hobby
- Many more… (see thesis)

It’s a tradeoff space, without one perfect answer

Goal: κ-Curve robustness, Spiro smoothness

- Is it possible?

This is perhaps the central research question I address.

Demo: spline.technology/demo

It's a design problem

- Don't expect a simple mathematical expression
- New approach: create a design tool for 2-parameter curve families
  - Arrange master curves on a grid
  - Exploit symmetries
  - With π/12 accuracy, need 49 masters
  - Each master is 2 cubic Béziers
- Design *exactly* the shape and behavior desired
- Encourage experimentation

Demo: spline.technology/demo/tuner.html

Straight-to-curve transitions

- Smooth curvature ramp up in U shapes
- Spiro used “one-way constraints”
- Good results but UX is difficult
  - Easy to get constraints wrong way

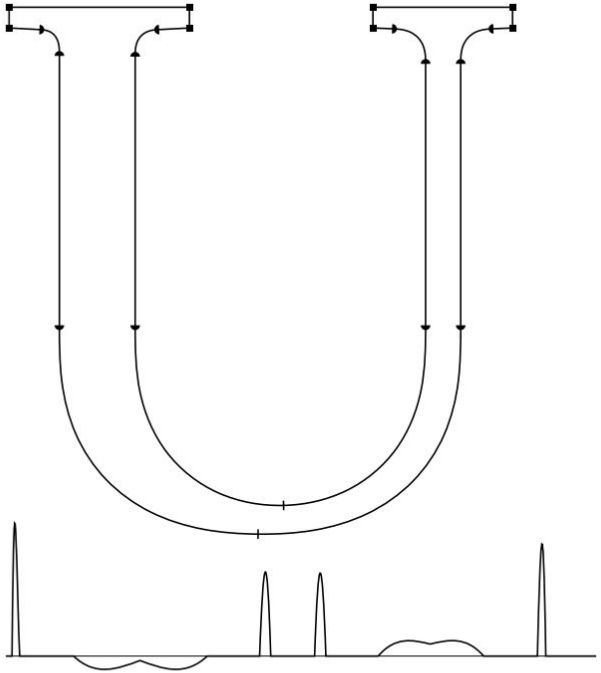
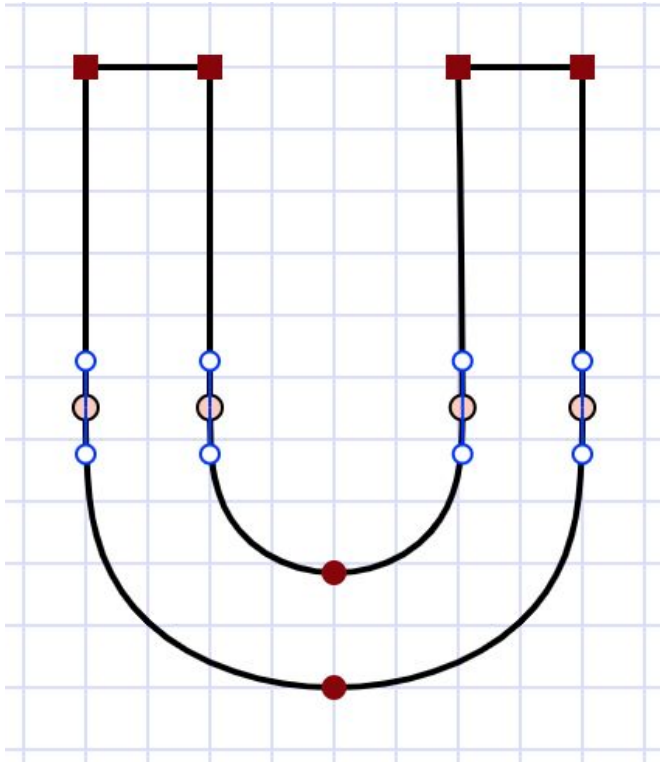


Figure 7.3. Bracketed serifs and straight-to-curve joins.

Straight-to-curve transitions, new approach

- Add *optional* tangent constraint to point
- UX is similar to Bézier curves
  - Length of control arm is currently not used, but could affect curvature
- Also useful for setting tangents at extrema
  - Spiro had no mechanism for this – important for fonts!



Prospects

- Will be used in the upcoming Runebender font editor
- More work needed on implementation
  - Solver needs to be 100% robust
  - Rust or (maybe) C
  - Smoother results with tangent constraints
- Will people prefer different tunings?
- What's the best UX for editing splines?
- Permissive license; I'd like to see this everywhere
- Work is funded by Google

Resources

- Code and writing: spline.technology
- Community: xi.zulipchat.com
- The font editor: linebender/runebender