Computer Graphics

Freeform Curves

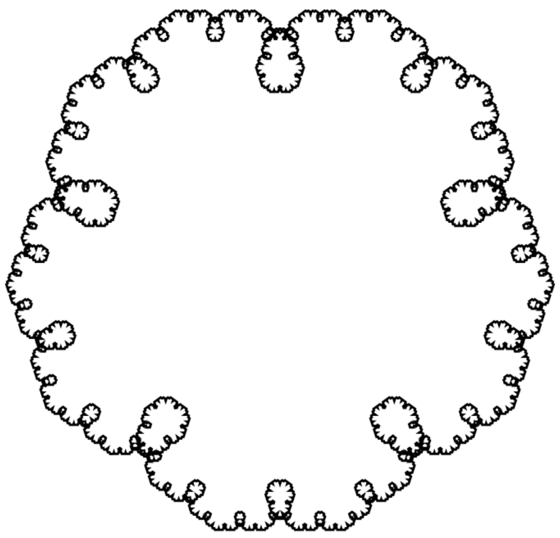
Mark Pauly

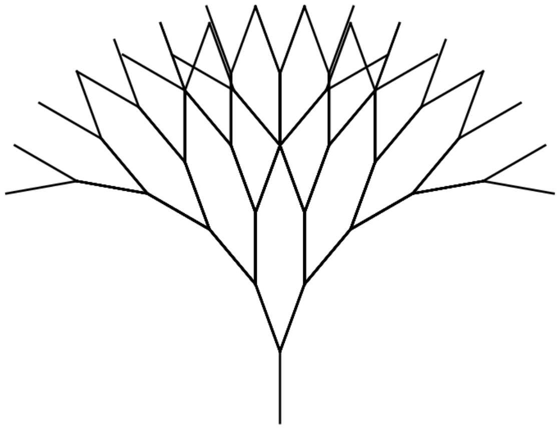
Geometric Computing Laboratory

Recap: L-System

Wikipedia

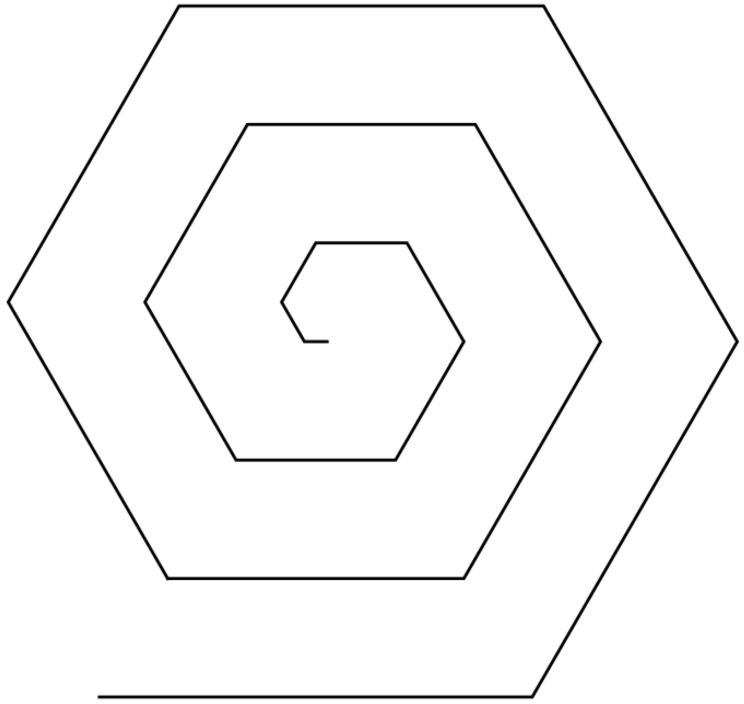
- An *L-system* is a string rewriting system (semi-Thue grammar)
  - Similar to context-free grammars, but rules executed in parallel
- L-System $\mathbf{G} = (V, \omega, P)$
  - Grammar on an alphabet of **symbols**, V, such as “F”, “+”, “-”.
  - **Production rules** P describe the replacement of a nonterminal symbol with a string of zero or more symbols.
  - Process is seeded with an **axiom** ω, an initial string
- Symbols can be interpreted as graphics commands (e.g. Turtle graphics)
- Branching can be implemeted using a stack.



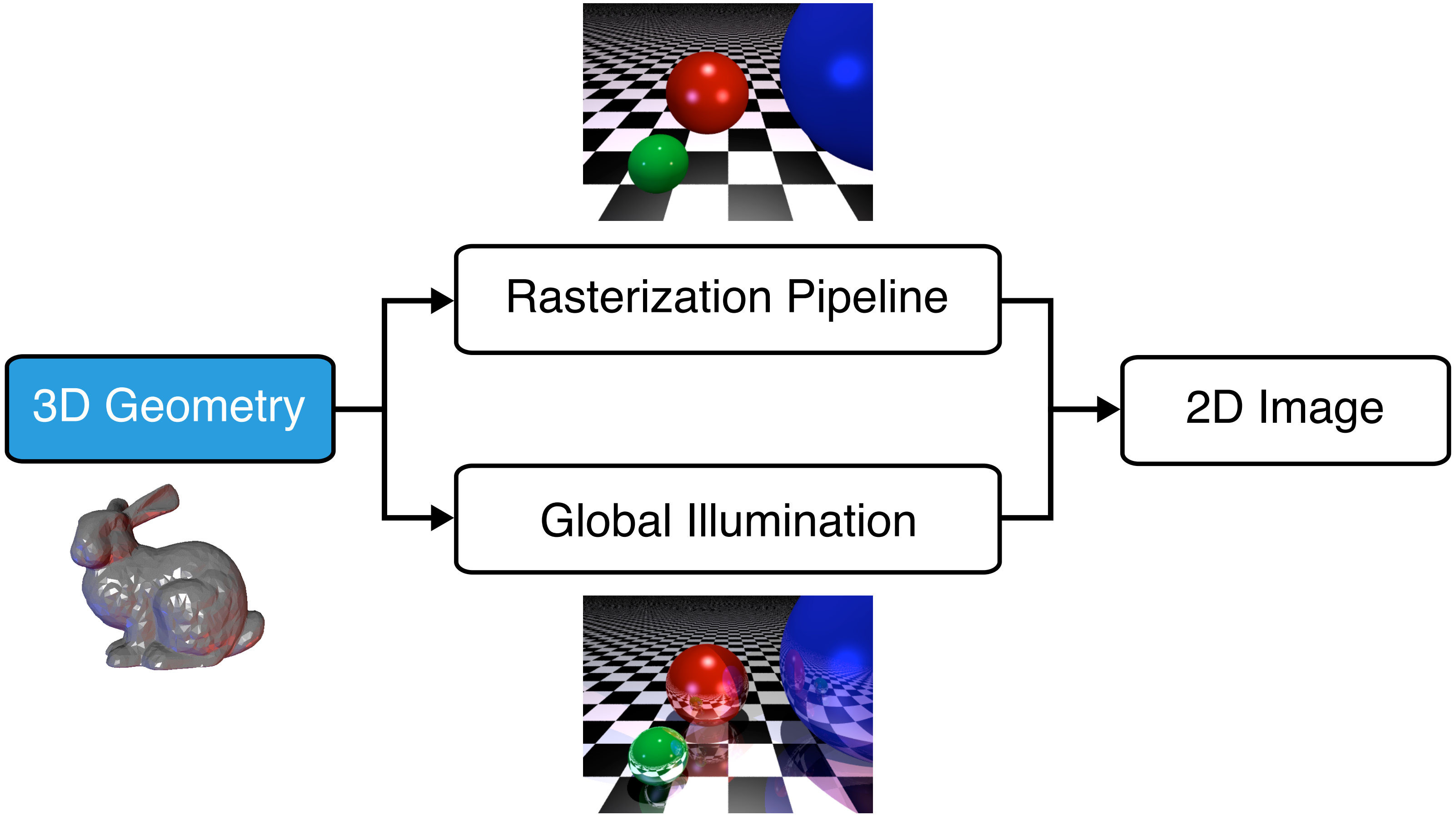


Inverse Problem: Spiral

- Which L-System creates this output?
  - define the axiom and the rule(s)



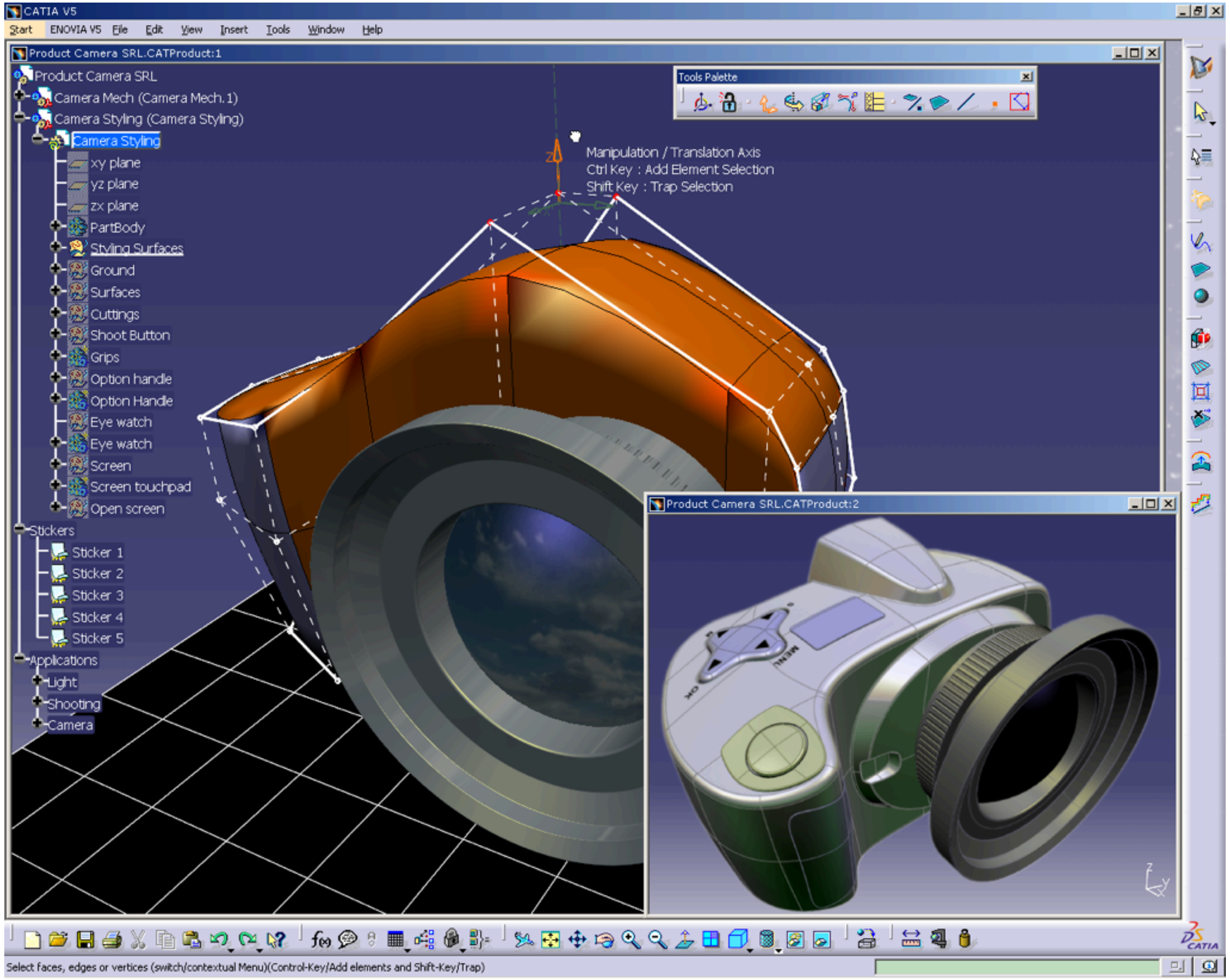
Overview


Rasterization Pipeline
3D Geometry
2D Image
Global Illumination

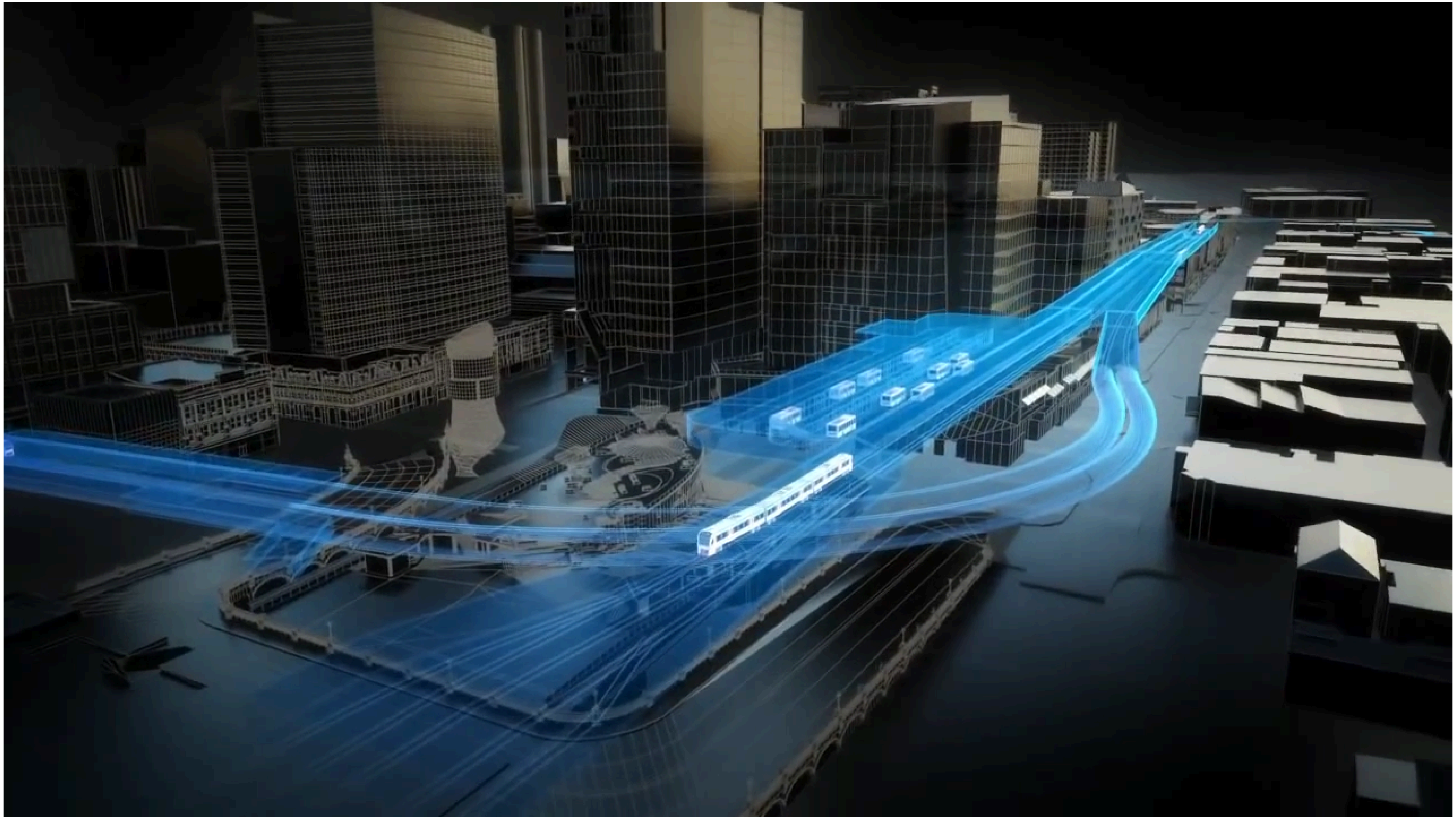
Geometric Modeling

- How did we model the scene so far?
  - Direct modeling: meshes (explicit), spheres, cylinders, planes (implicit)
  - Procedural modeling: L-Systems, Terrains, etc.
- Now: Brief introduction to freeform curves & splines
  - Polynomial Bezier curves and surfaces
  - Piecewise polynomial spline curves and surfaces
  - Freeform deformation

Example: Dassault's CATIA

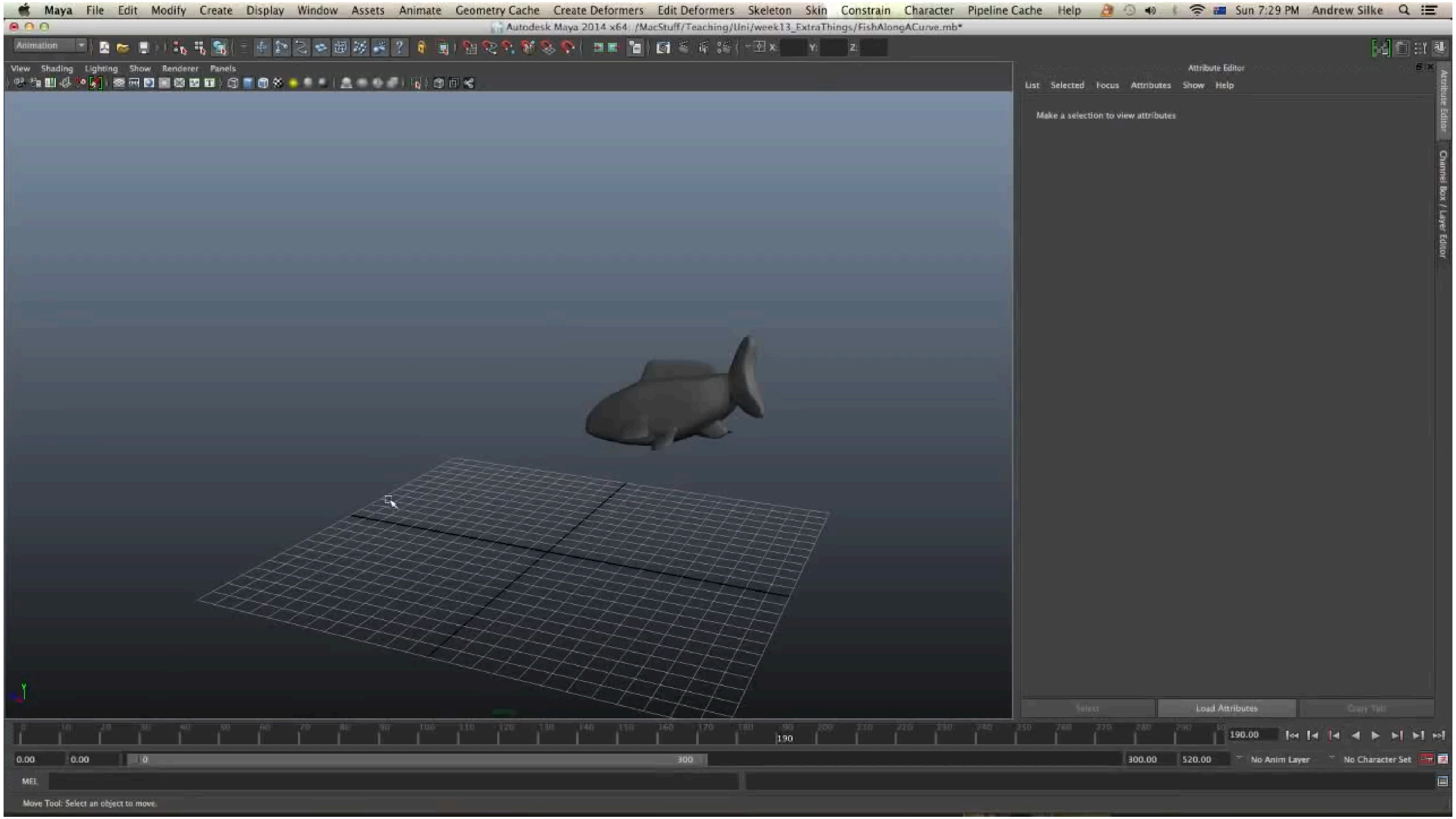


Example: Camera Path



Source

Example: Animation Curves

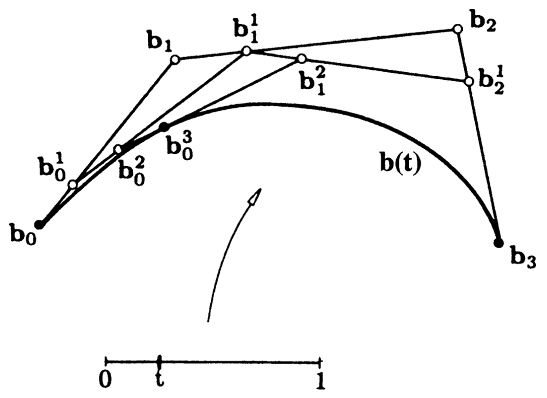


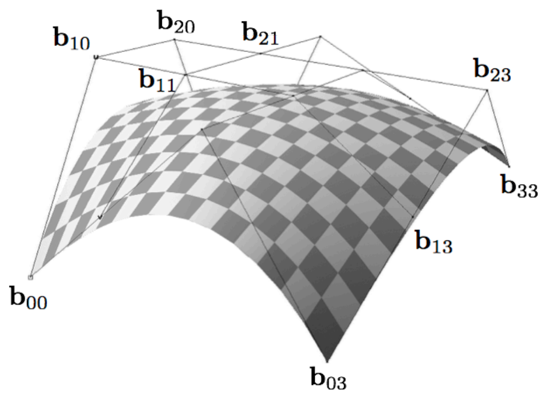
Requirements

- Which properties are important for a geometry representation?
  - Approximation power
  - Efficient evaluation of positions & derivatives
  - Ease of manipulation
  - Ease of implementation

Overview

1. Freeform Curves
2. Freeform Surfaces
3. Freeform Deformation





Freeform Curves

How to make LOVE?

getdigital

But how to draw a heart?

We have to understand parametric space curves!

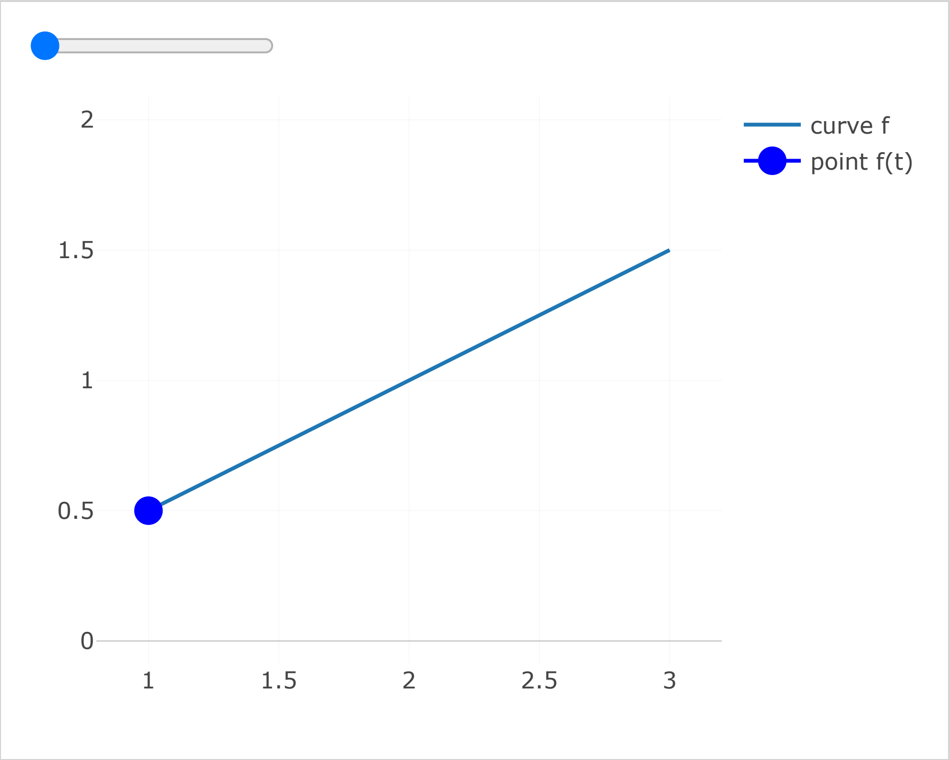
Parametric Curve Representation

- Parametric representation $\mathbf{x}\colon [a, b] \subset \mathbb{R} \to \mathbb{R}^3$ (or $\mathbb{R}^2$)

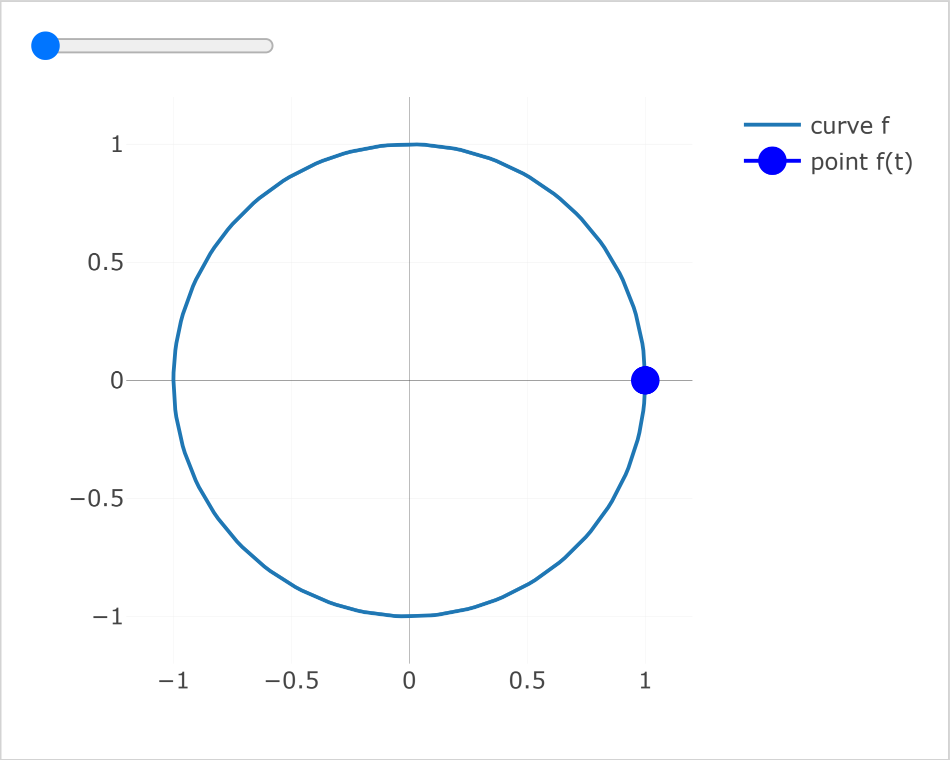
$$\mathbf{x}(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix}$$

- Curve is defined as the *image* of the interval $[a, b]$ under the continuous parameterization function $\mathbf{x}$.

Examples

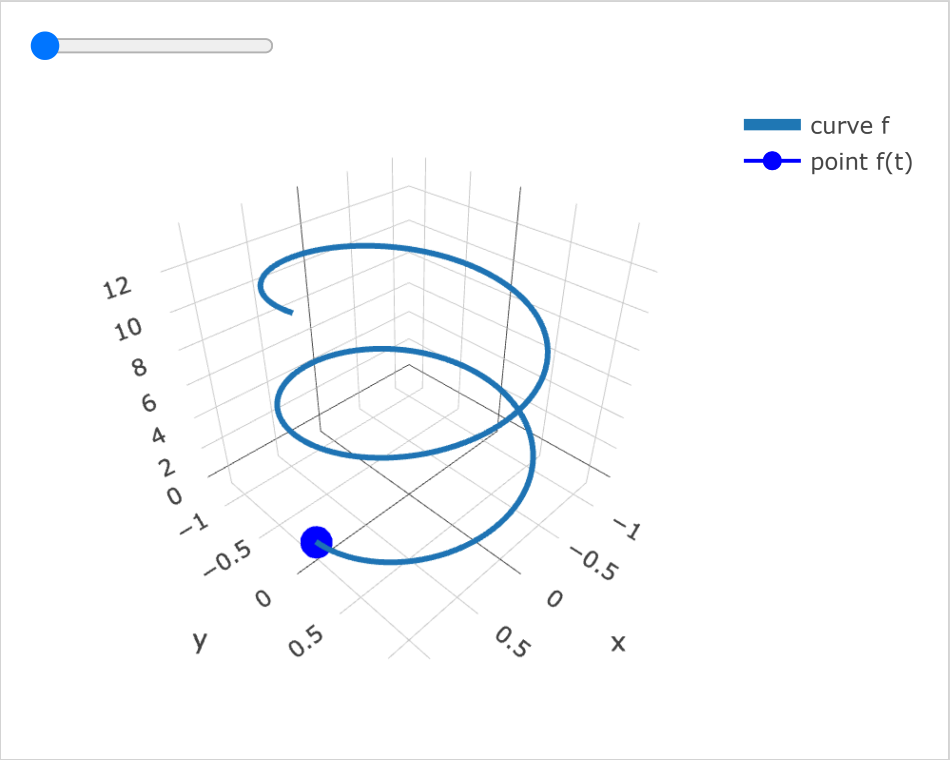


$$\mathbf{f}(t) = \begin{pmatrix} 1 \\ 0.5 \end{pmatrix} + t \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

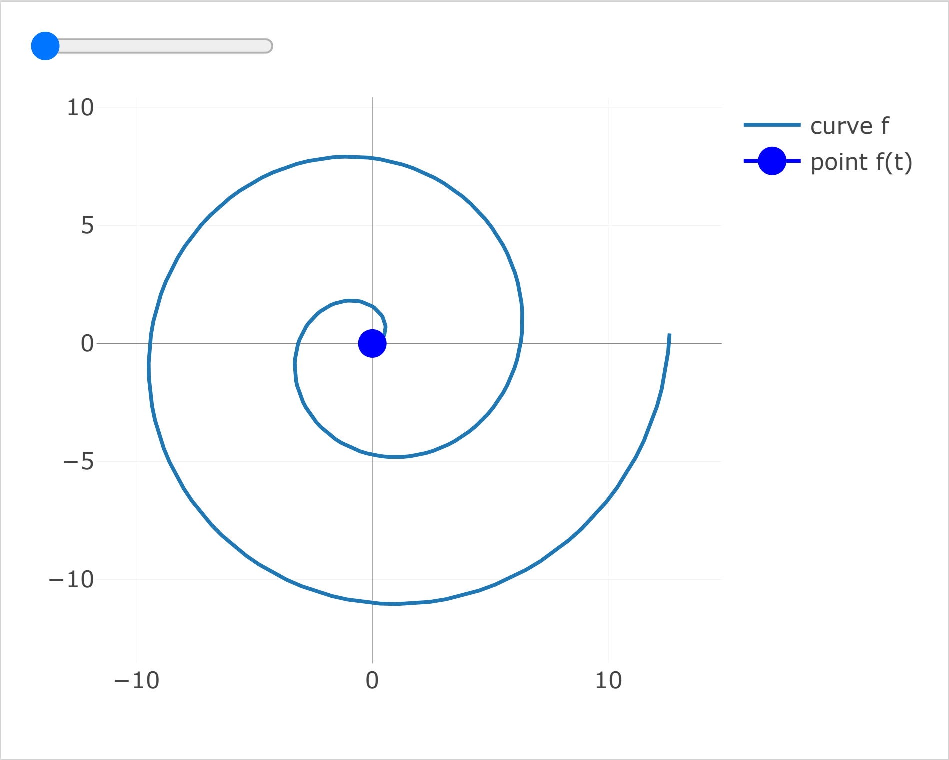


$$\mathbf{f}(t) = \begin{pmatrix} \cos(t) \\ \sin(t) \end{pmatrix}$$

Examples



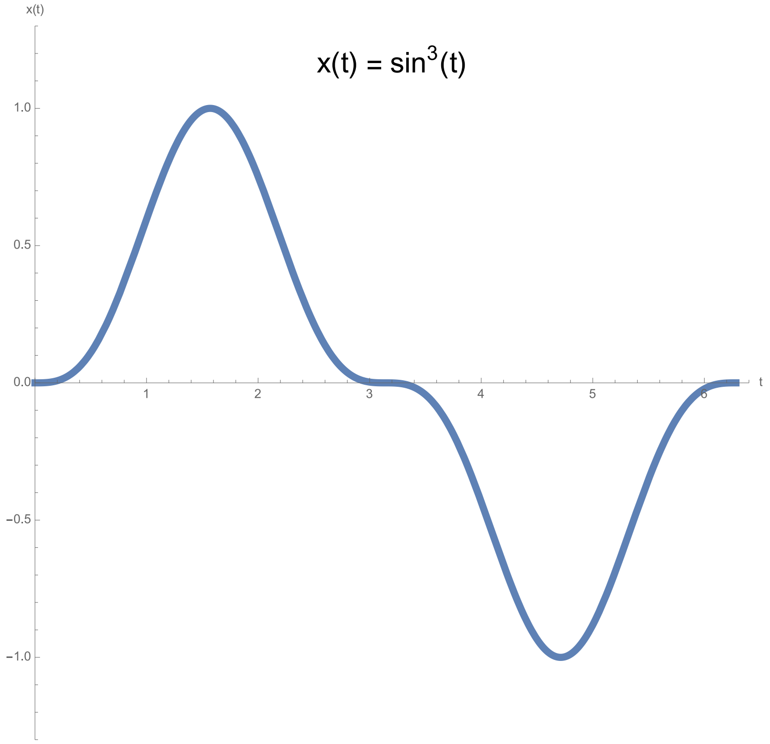
$$\mathbf{f}(t) = \begin{pmatrix} \cos(t) \\ \sin(t) \\ t \end{pmatrix}$$

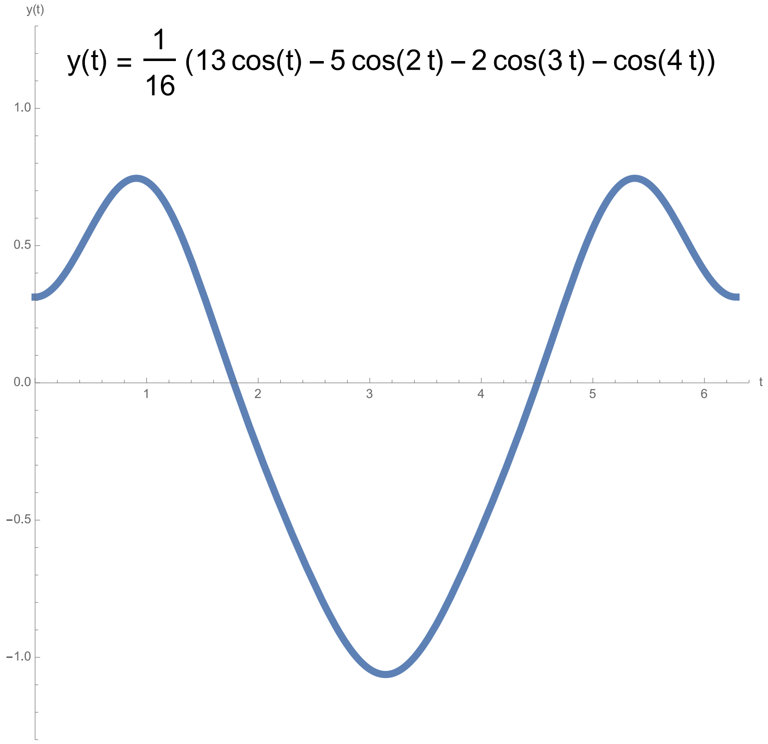


$$\mathbf{f}(t) = \begin{pmatrix} t\cos(t) \\ t\sin(t) \end{pmatrix}$$

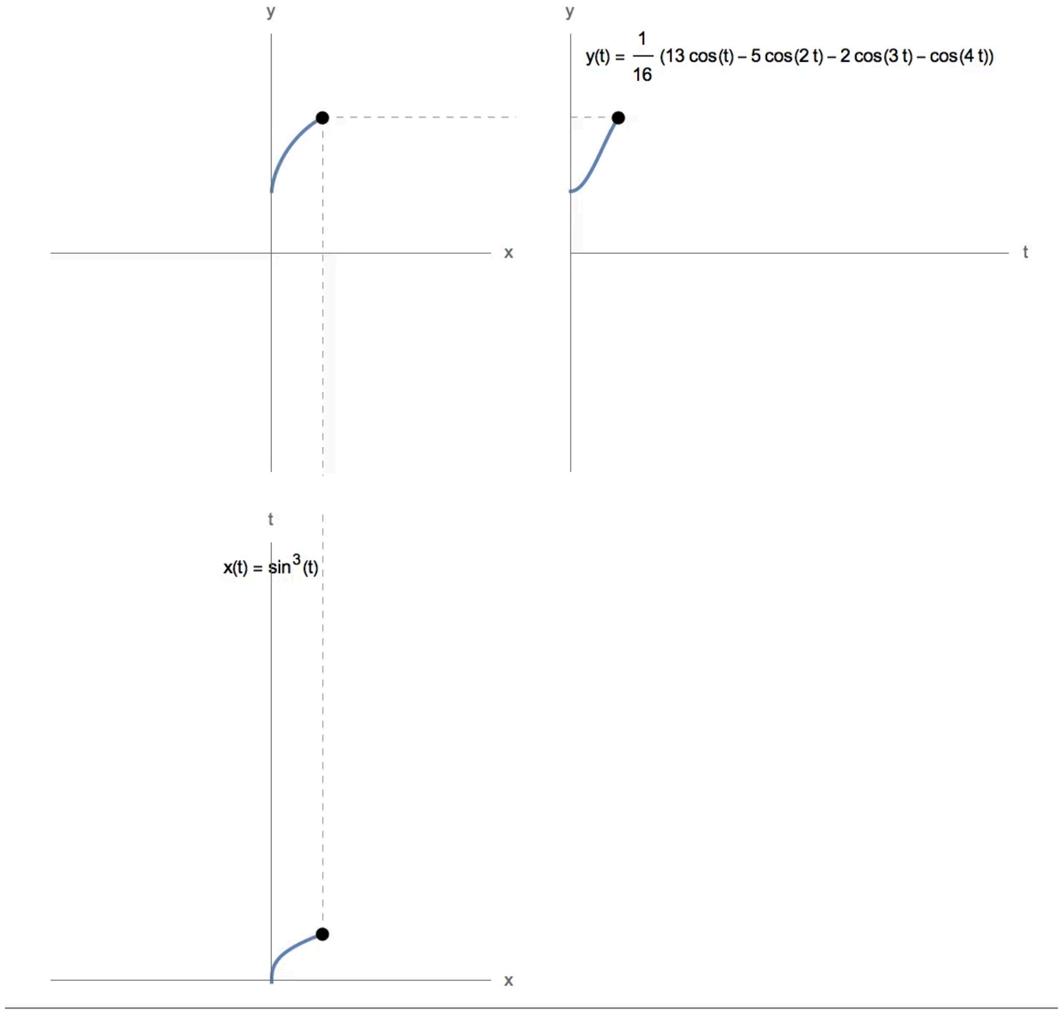
Guess the shape of the curve!

$$\mathbf{x}(t) = \begin{pmatrix} \sin^3(t) \\ \frac{1}{16}(13\cos(t) - 5\cos(2t) - 2\cos(3t) - \cos(4t)) \end{pmatrix}$$





Guess the shape of the curve!



Tangent Vector

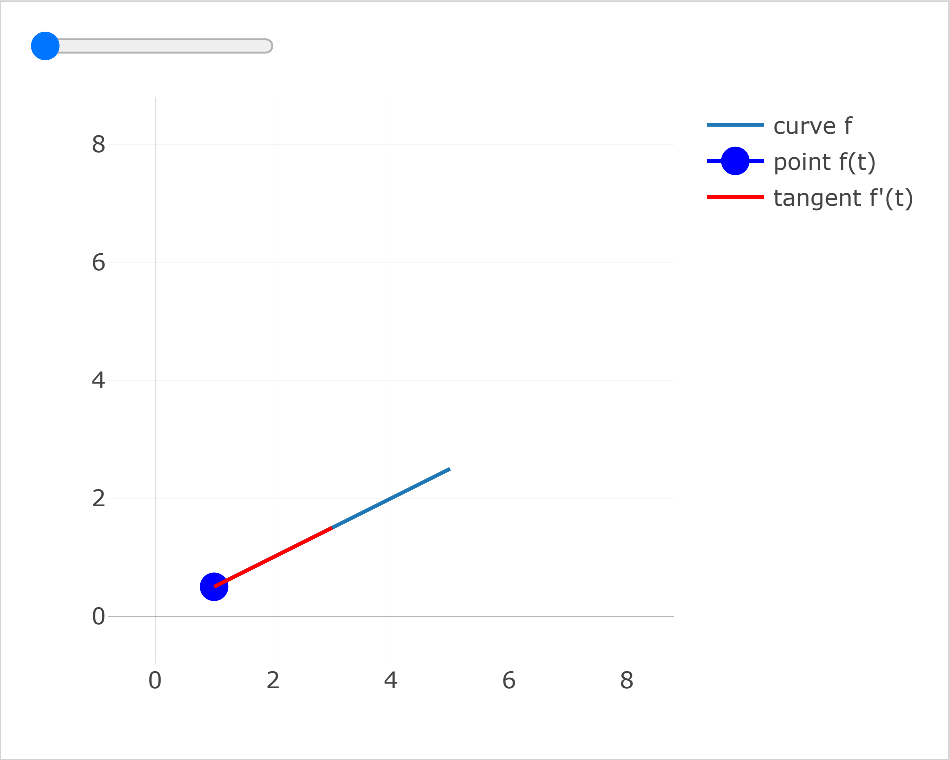
- Parametric curve representation

$$\mathbf{x}(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix}$$

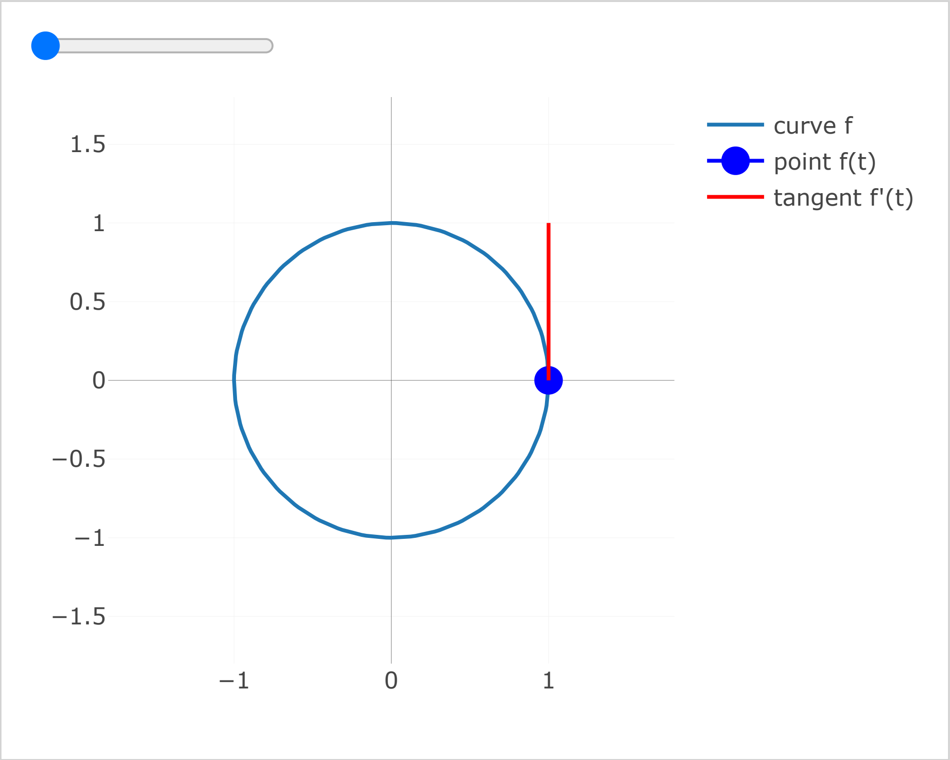
- First derivative defines the *tangent vector*

$$\mathbf{t} = \mathbf{x}'(t) := \frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \begin{pmatrix} \mathrm{d}x(t)/\mathrm{d}t \\ \mathrm{d}y(t)/\mathrm{d}t \\ \mathrm{d}z(t)/\mathrm{d}t \end{pmatrix}$$

Examples

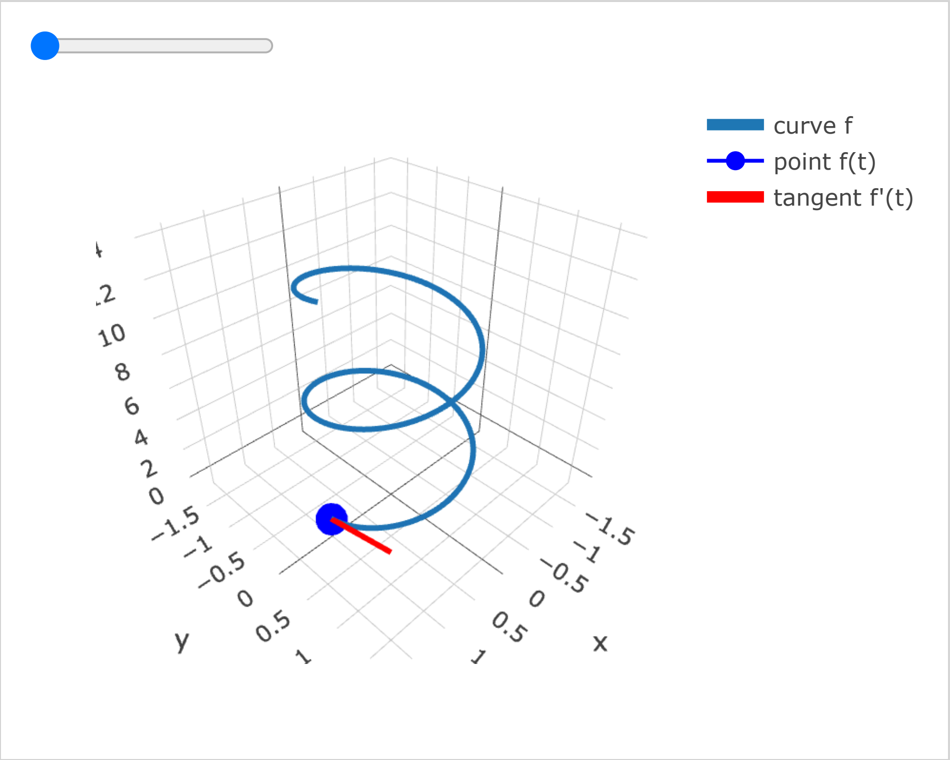


$$\mathbf{f}(t) = \begin{pmatrix} 1 \\ 0.5 \end{pmatrix} + t \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

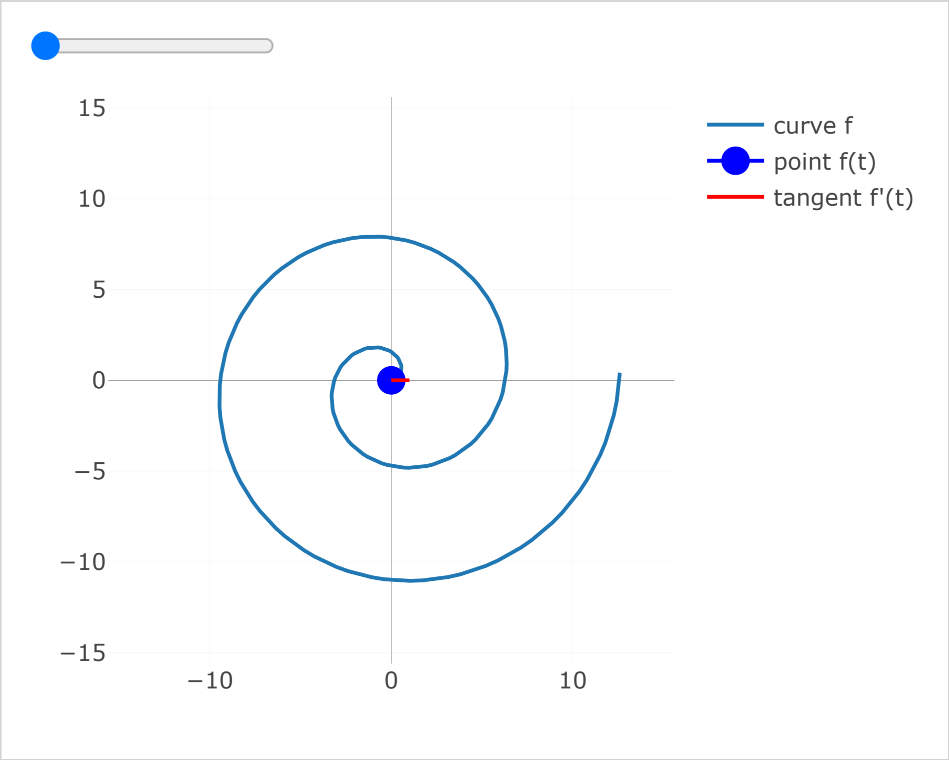


$$\mathbf{f}(t) = \begin{pmatrix} \cos(t) \\ \sin(t) \end{pmatrix}$$

Examples



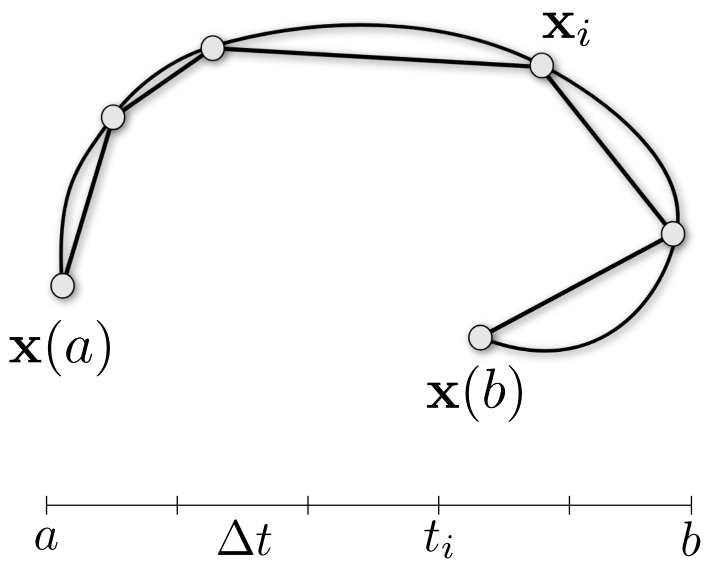
$$\mathbf{f}(t) = \begin{pmatrix} \cos(t) \\ \sin(t) \\ t \end{pmatrix}$$



$$\mathbf{f}(t) = \begin{pmatrix} t\cos(t) \\ t\sin(t) \end{pmatrix}$$

Discrete Curves

- Approximate the curve by a polygon, e.g. for rendering
  1. Sample parameter interval: $t_i = a + i\Delta t$
  2. Sample curve: $\mathbf{x}_i = \mathbf{x}(t_i)$
  3. Connect samples by polygon



Polynomial Curves

- Let's model curves as polynomials of degree n

$$\mathbf{x}(t) = \sum_{i=0}^{n} \mathbf{b}_i \, \phi_i(t) \ \in \Pi^n$$

  - Vector-valued coefficients $\mathbf{b}_i \in \mathbb{R}^3$
  - Scalar-valued basis polynomials $\phi_i \colon \mathbb{R} \to \mathbb{R}$
  - $\{\phi_0, \ldots, \phi_n\}$ span space of degree n polynomials

Which basis of the space of polynomials should we use?

Polynomial Curves

Which properties does the monomial basis provide?

$$\mathbf{x}(t) \;=\; \sum_{i=0}^{n} \mathbf{b}_i \, t^i$$

✓ Approximation power

- The Weierstrass approximation theorem guarantees that polynomials have nice approximation power.

✓ Efficient evaluation of positions & derivatives

- With only addition and multiplication, polynomials can be efficiently evaluated

✓ Ease of implementation

- You only need addition and multiplication, and that is very easy.

✗ Ease of manipulation

- Because the monomials do **not** add up to one, i.e., $\sum_i t_i \neq 1$, the sum of points $\mathbf{x}(t)$ is not an affine combination and hence geometrically does not make sense. The coefficients $\mathbf{b}_i$ do not have a geometric interpretations.

Monomial Basis

Check requirements for geometry representation:

- ✓ Approximation power
- ✓ Efficient evaluation of positions & derivatives
- ✗ Ease of manipulation
- ✓ Ease of implementation

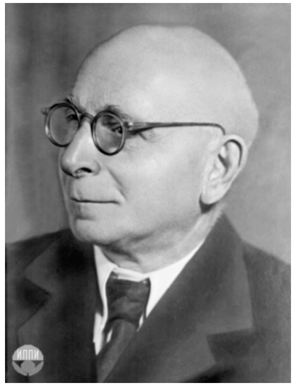
Let's find a better basis of Π^n!

Bernstein Polynomials

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad 0 \leq i \leq n$$

$$\binom{n}{i} = \begin{cases} \frac{n!}{i!(n-i)!} & \text{if } 0 \leq i \leq n, \\ 0 & \text{otherwise.} \end{cases}$$

code-3b19eeca.gnuplot.svg



Sergei Bernstein, 1880–1968

Bernstein Polynomials

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

code-3b19eeca.gnuplot.svg

- Basis: $\{B_0^n, \ldots, B_n^n\}$ are a basis for Π^n
- Non-negativity: $B_i^n(t) \geq 0$ for $t \in [0, 1]$
- Endpoints: $B_i^n(0) = \delta_{i,0}$ and $B_i^n(1) = \delta_{i,n}$
- Symmetry: $B_i^n(t) = B_{n-i}^n(1-t)$
- Maximum: $B_i^n(t)$ has maximum at $t = i/n$.
- Partition of unity: $\sum_{i=0}^n B_i^n(t) = 1$

Partition of Unity

How can we show that $\sum_{i=0}^{n} B_i^n(t) = 1$?