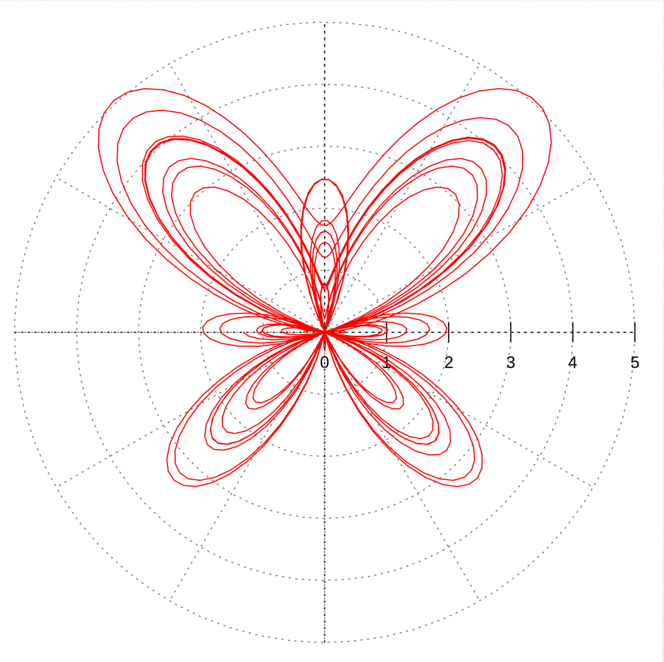


Dr. Sarah Abraham
University of Texas at Austin
Computer Science Department

Non-linear Motion

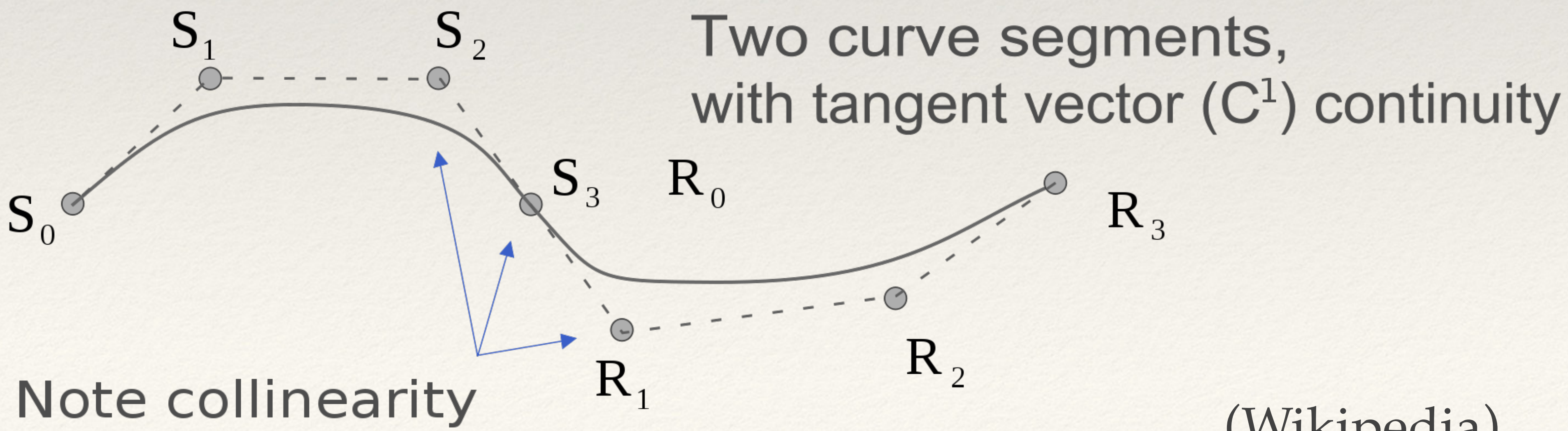
Elements of Graphics
CS324e

Interpolation

- Linear interpolation is a simple form of interpolation
  - Can only move along a line
  - Distance traveled between steps is constant
- Cosine interpolation provides smooth curves between points, but it does not allow for “true” continuity

What is Continuity?

- Defines level of smoothness along a curve
- Types of continuity:
  - C0: curves are joined
  - C1: first derivatives of curves are continuous
  - C2: second derivatives of curves are continuous

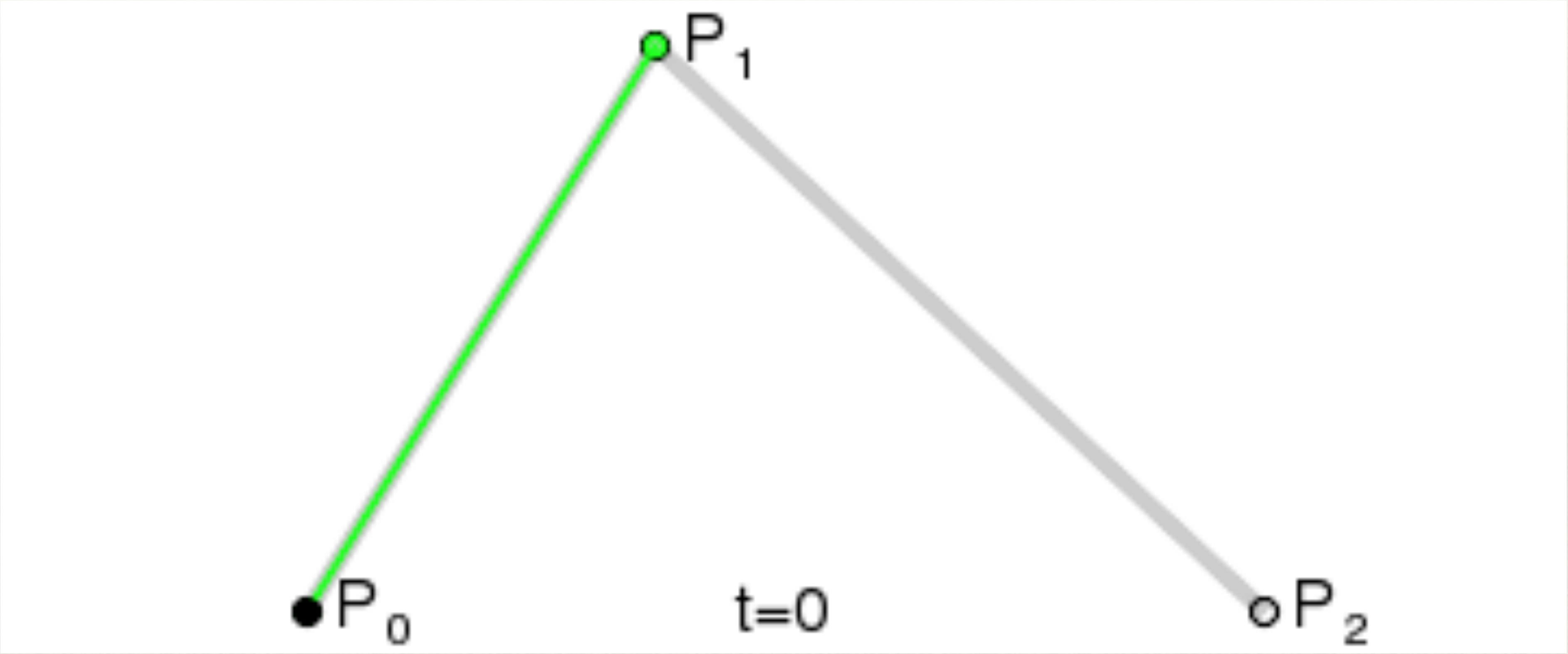


(Wikipedia)

Higher Continuity

- Higher-order continuities have advantages in both geometric modeling and in animation
- Some overhead, since additional points of data are required
- Splines are piecewise polynomials (multiple, connected lower-degree polynomial functions) with high continuity

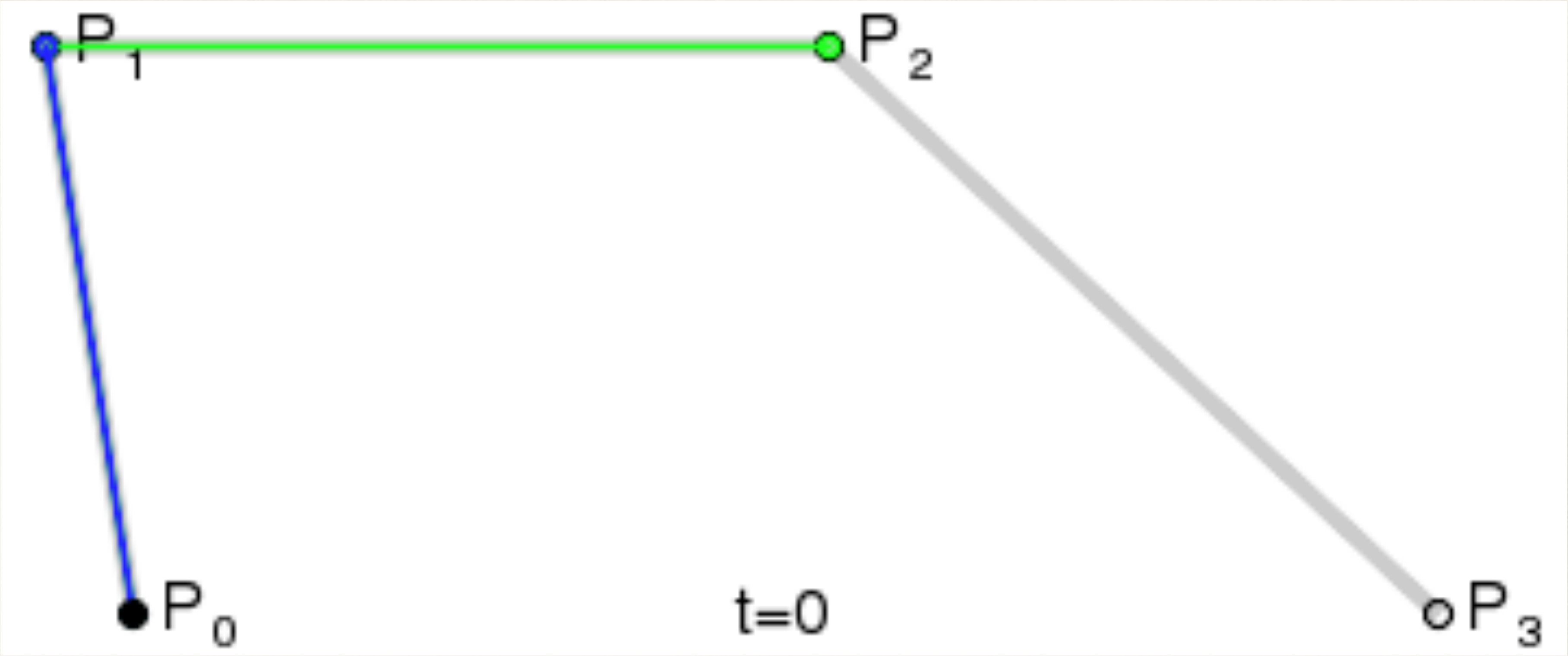
Quadratic Bezier Curve



Question

- How does the quadratic Bezier curve interpolate between points?

Cubic Bezier Curve

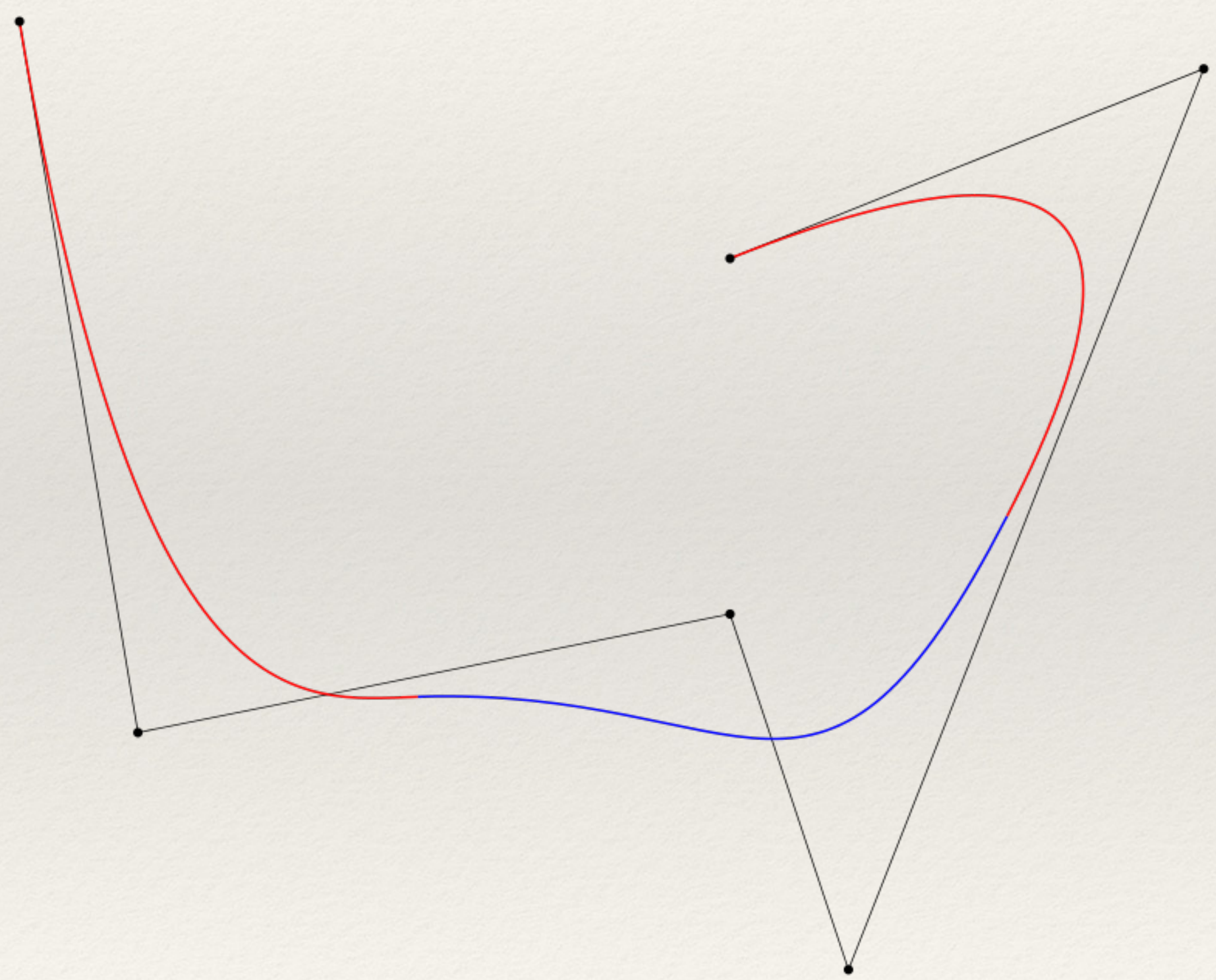


Question

- How does the cubic Bezier curve interpolate between points?

Combining Bezier Curves

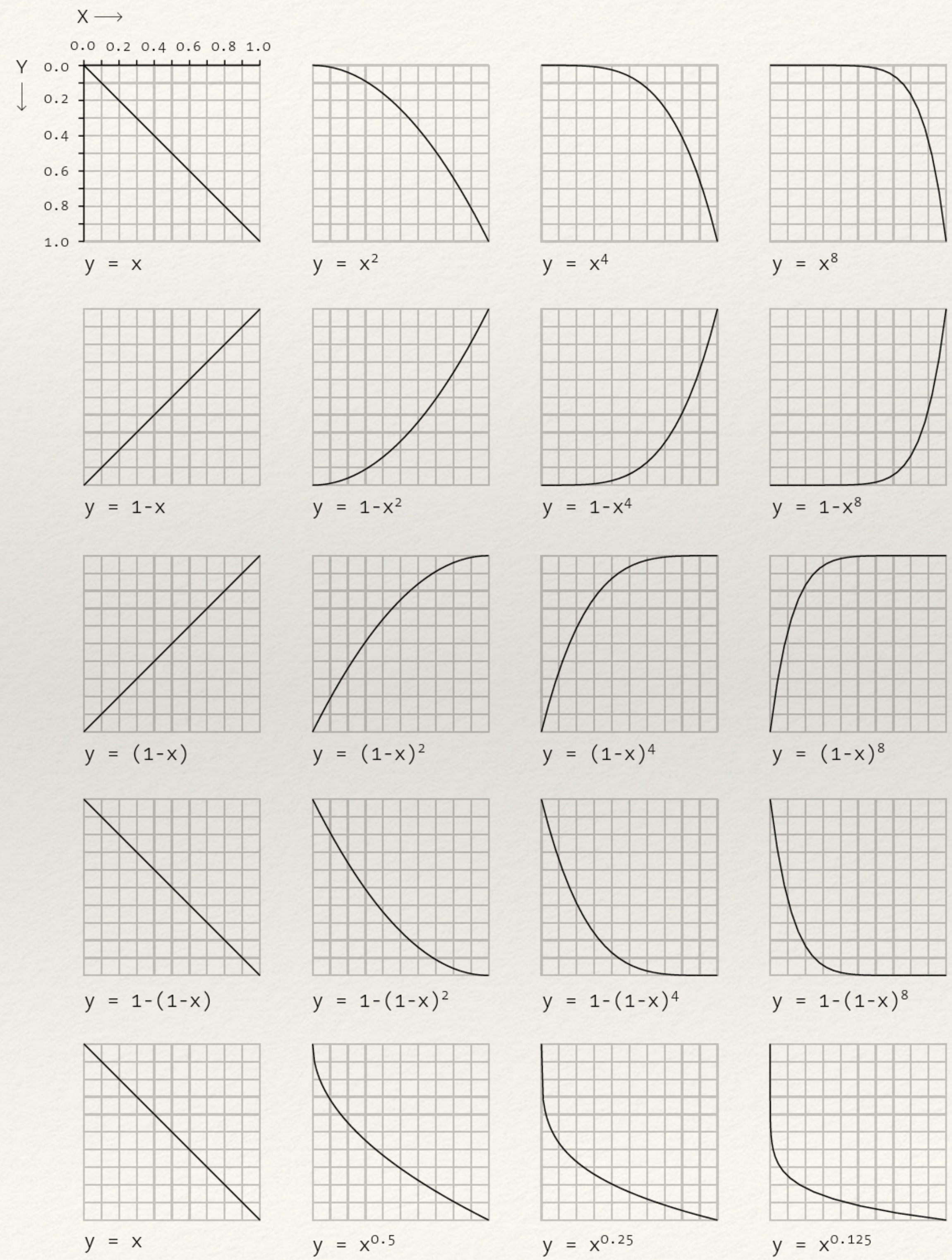
Bezier curves can be joined to form a B-spline!



Sequence of Bezier curves and control points that form a B-spline (Wikipedia)

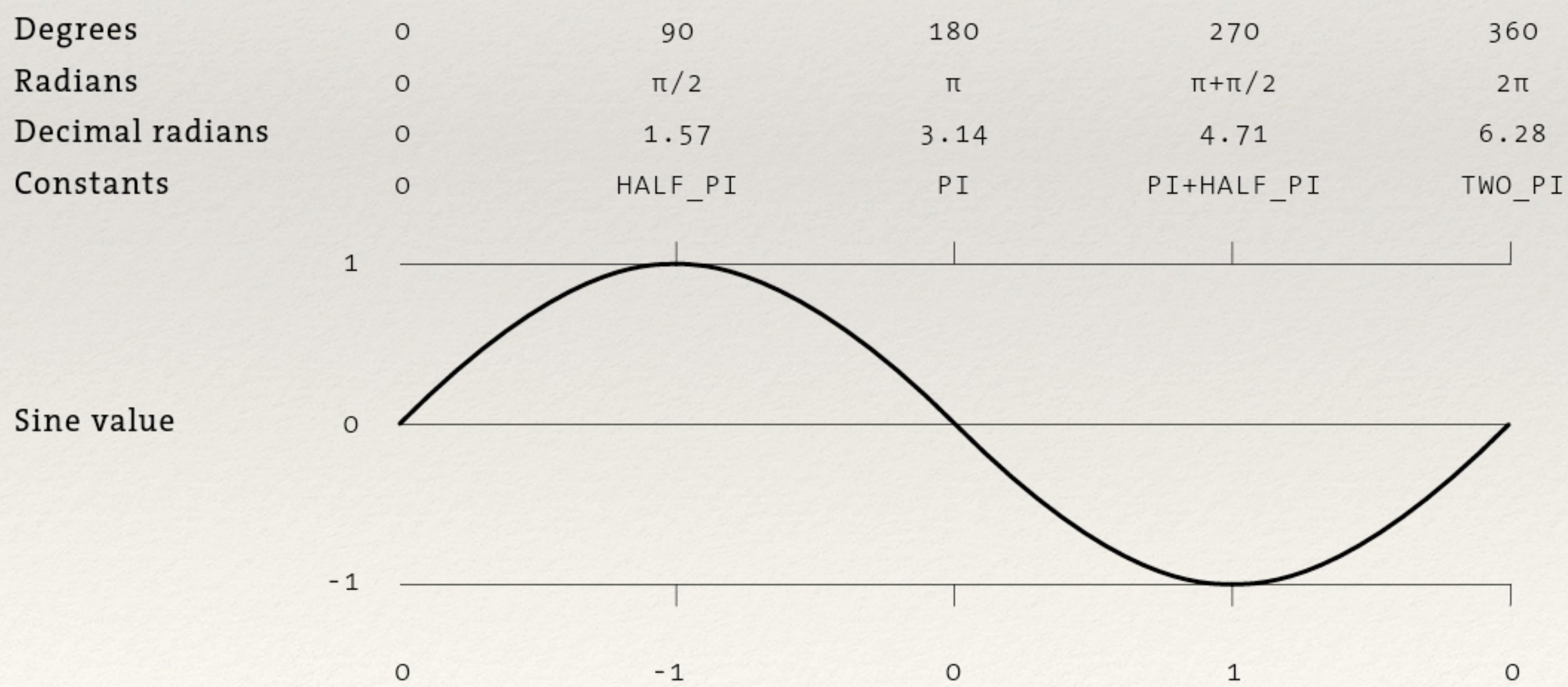
Other Interpolation Functions

- Interpolation calculates intermediate values between targets
- Interpolation along arbitrary functions generates a range of behaviors



Sine Waves

- Sine (and cosine) equations model a periodic relationship between radians and sine/cosine value
- Sine waves have an angle (Θ), amplitude (*a*) and phase (*p*)



Using Sine

- sin(ϴ) outputs a value between -1 and 1 based on ϴ (between 0 and 2PI)
- Amplitude *a* magnifies the value of the sine peak
- Phase *p* controls where the oscillation cycle begins
- Sine can control rate of motion (frequency) as well as direction of motion

Question

- If $\sin(\Theta)$ and $\cos(\Theta)$ are applied to the x and y position of an object respectively, what will that object's motion be?

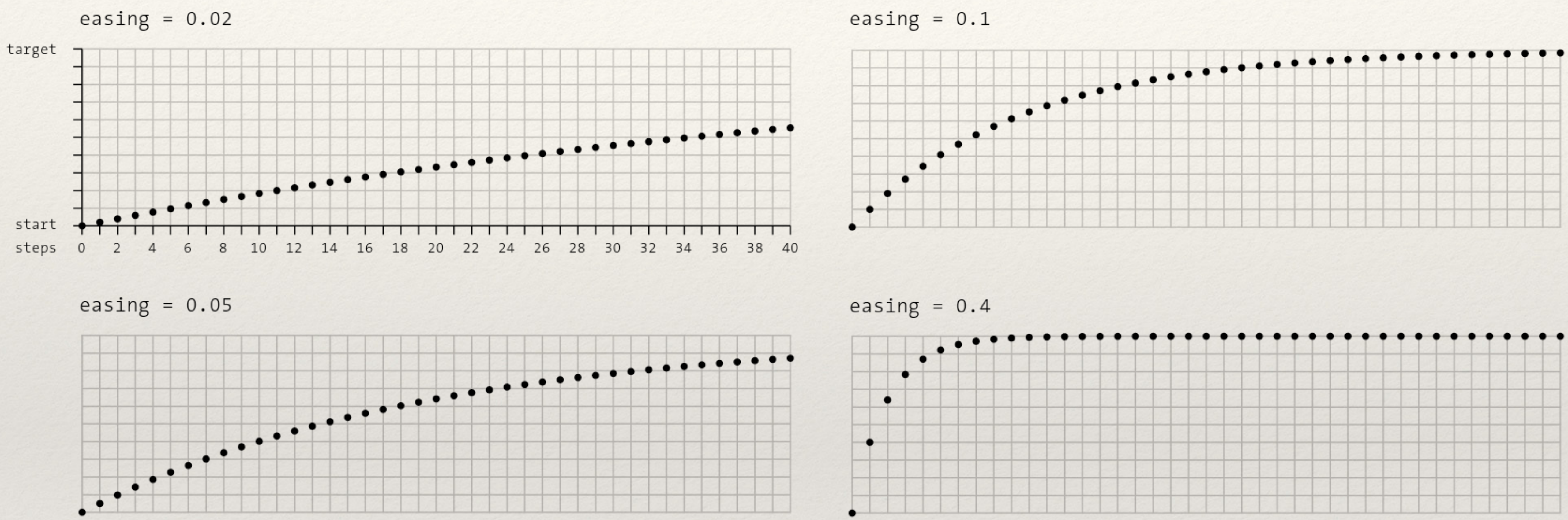
Easing

- Easing allows movement between two values at nonlinear increments
  - Objects can accelerate/decelerate as they approach the target
- Equation determines the fraction of the distance between the object's current and target positions that the object moves

Ease-out Example

```
float x = 0.0;
float easing = 0.05;
float targetX = 400;
void setup() {size(500, 500);}
void draw() {
  x += (targetX - x) * easing;
  ellipse(x, 250, 50, 50);
}
```

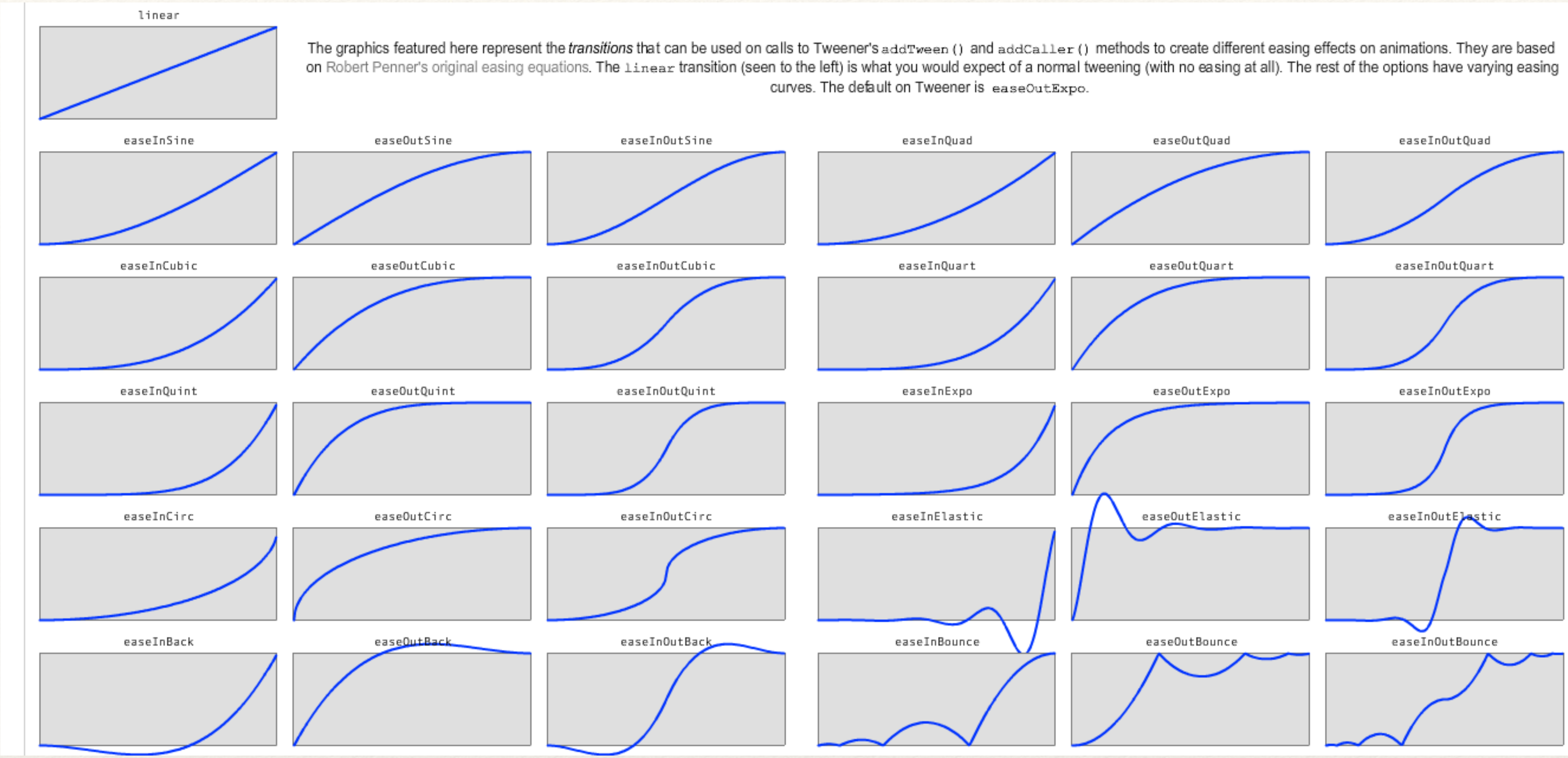
Ease-out as a Function



- We can also think of ease-out as a *lerp* operation over non-even time increments

Types of Ease Functions

- Robert Penner defined a range of easing equations: http://gizma.com/easing/
- These equations are based on change in value over time
  - Ease in
  - Ease out
  - Ease in/out
- Basic functions are linear, quadratic, cubic, quartic and quintic equations



(https://code.google.com/archive/p/tweener/)

Hands-on: Using Non-linear Motion

- Today's activities:
  1. Use sin(Θ) to oscillate a ball back and forth from a point
  2. Now use sin(Θ) to circle a ball around a point
  3. Rewrite the ease-out function using the `lerp` method
  4. Create an ease-in function using the `lerp` method. Note that you will need to track the total distance between start and end position as well as the current distance to the end position to do this