Announcements

- Anyone still on the wait list?
- Assignment 1 out on Thursday

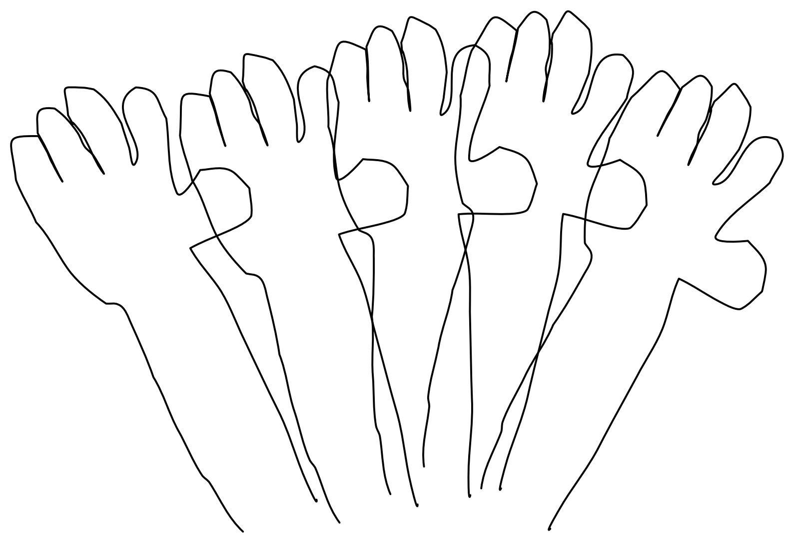
Keyframing

COMPUTER ANIMATION

15-497/15-861

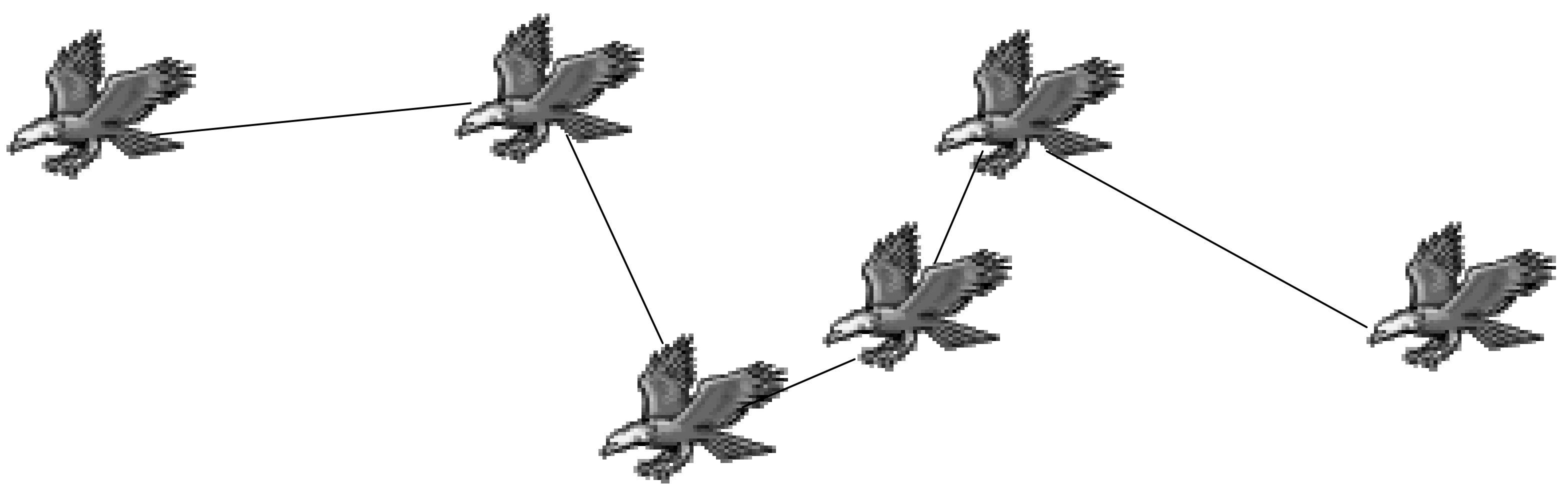
Keyframing in 2D

- Highly skilled animator draws the important, or key frames
- Less skilled (lower paid) animator draws the in-between frames



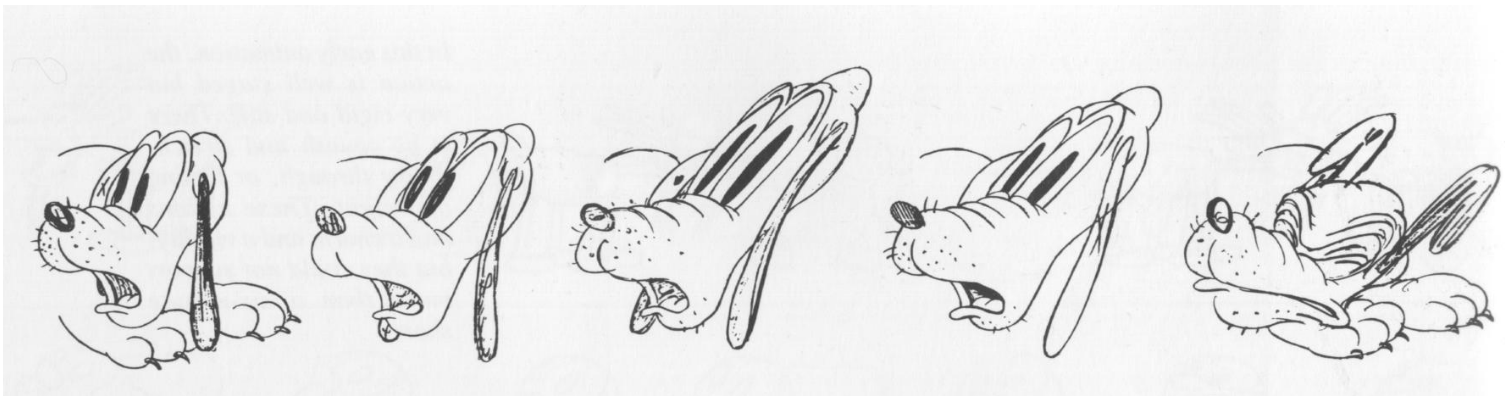
Keyframing in 3D

- Animator specifies the important key frames
- Computer generates the in-betweens automatically using interpolation
- Rigid body motion isn’t nearly enough—even for this sprite



What is a key?

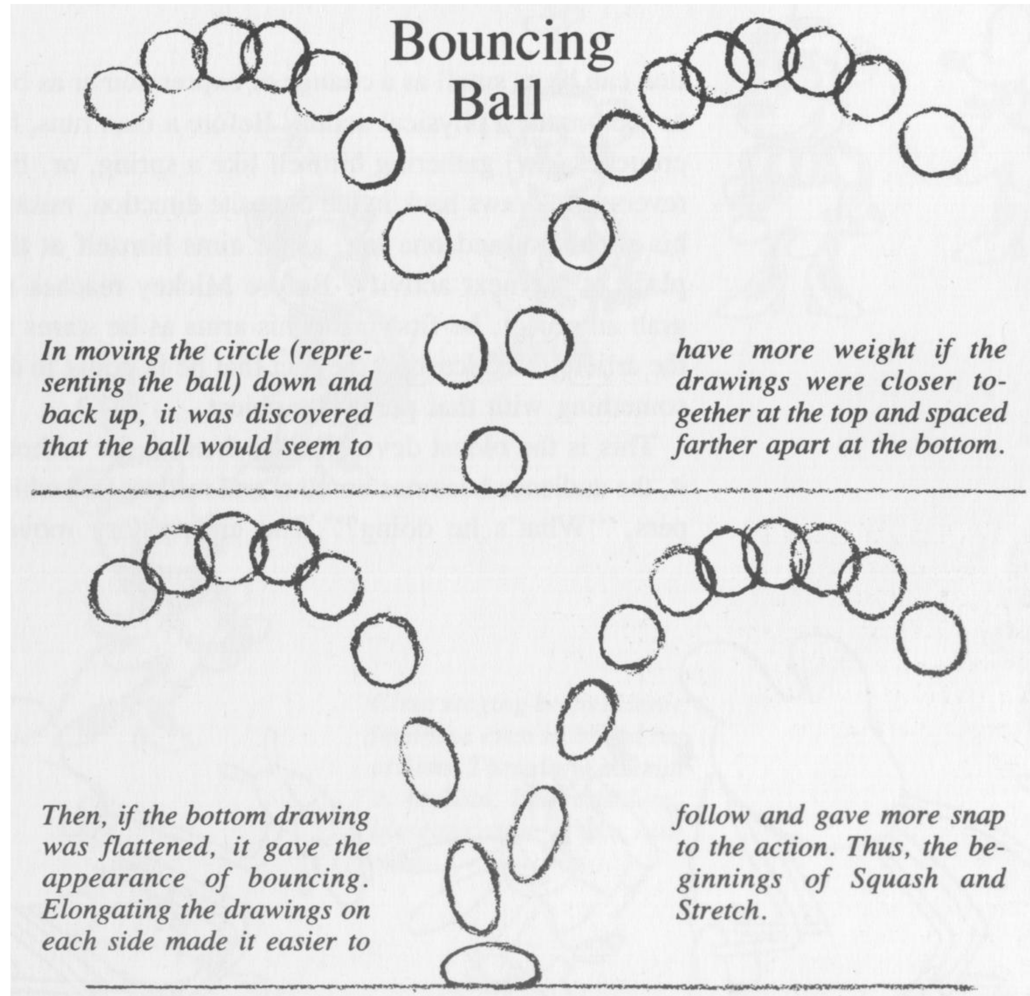
- Hard to interpolate hand-drawn images
  - Computers don’t help much



- The situation is different in computer animation:
  - Each keyframe is a defined by a bunch of parameters (state)
  - Sequence of keyframes = points in high-dimensional state space
- Computer inbetweening interpolates these points

What is a key?

- For a bouncing ball?
  - –Position in 3D
  - –Orientation?
  - –Squishedness?



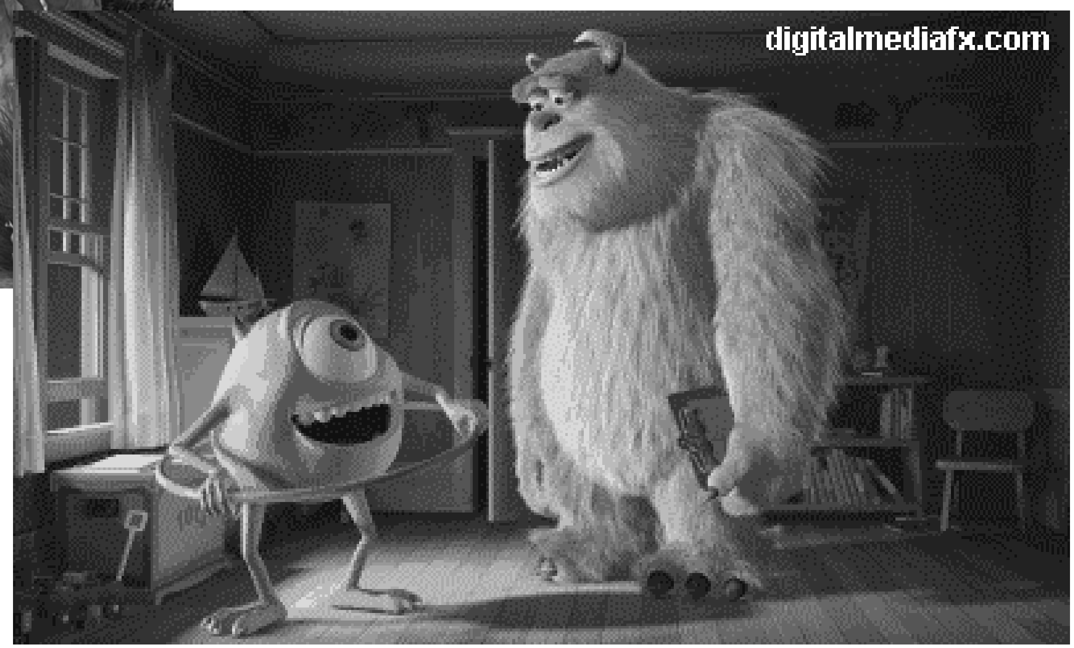
What is a key?

- For a monster?
  - –Position and orientation in 3D
  - –Joint angles of the hierarchy
  - –Deformations?
  - –Facial features
  - –Hair/fur???
  - –Clothing???
- Scene elements?
  - –Lights
  - –Camera

© 2001 Disney/Pixar

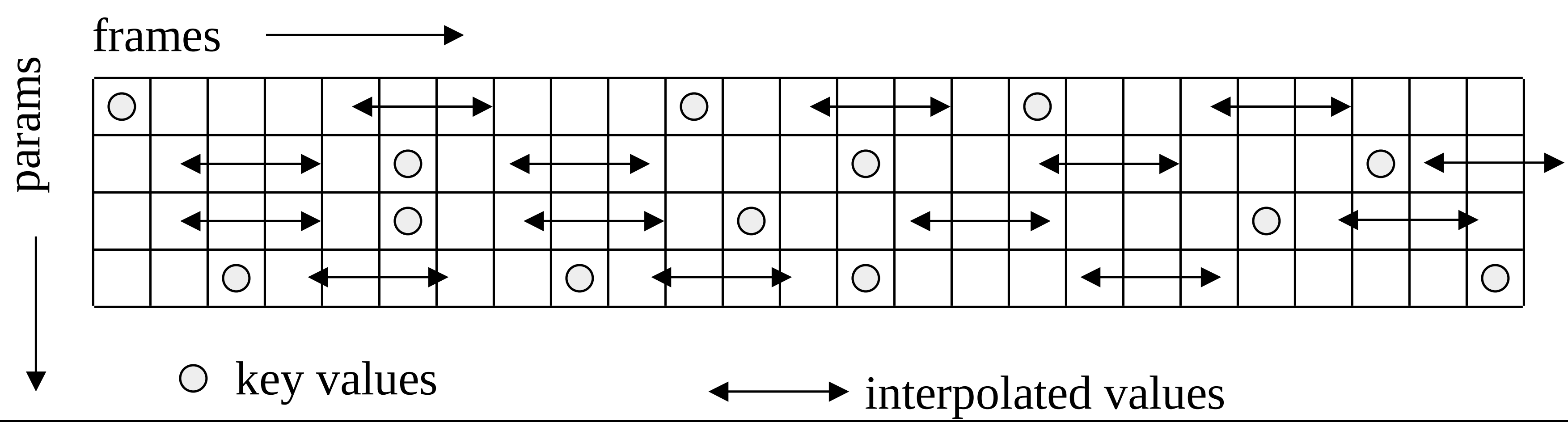
Monster trailers…

Digression: Should Monsters or Shrek win the Oscar?



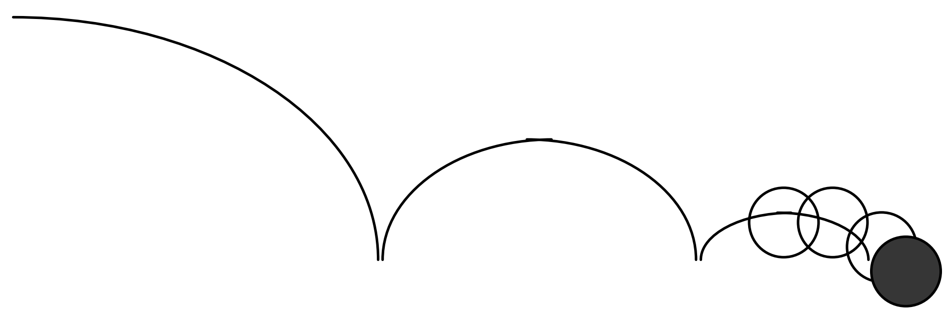
Keyframing Basics

- Despite the name, there aren’t really keyframes, *per se*.
- For each variable, specify its value at the “important” frames. Not all variables need agree about which frames are important.
- Hence, *key values* rather than key frames
- Create path for each parameter by interpolating key values

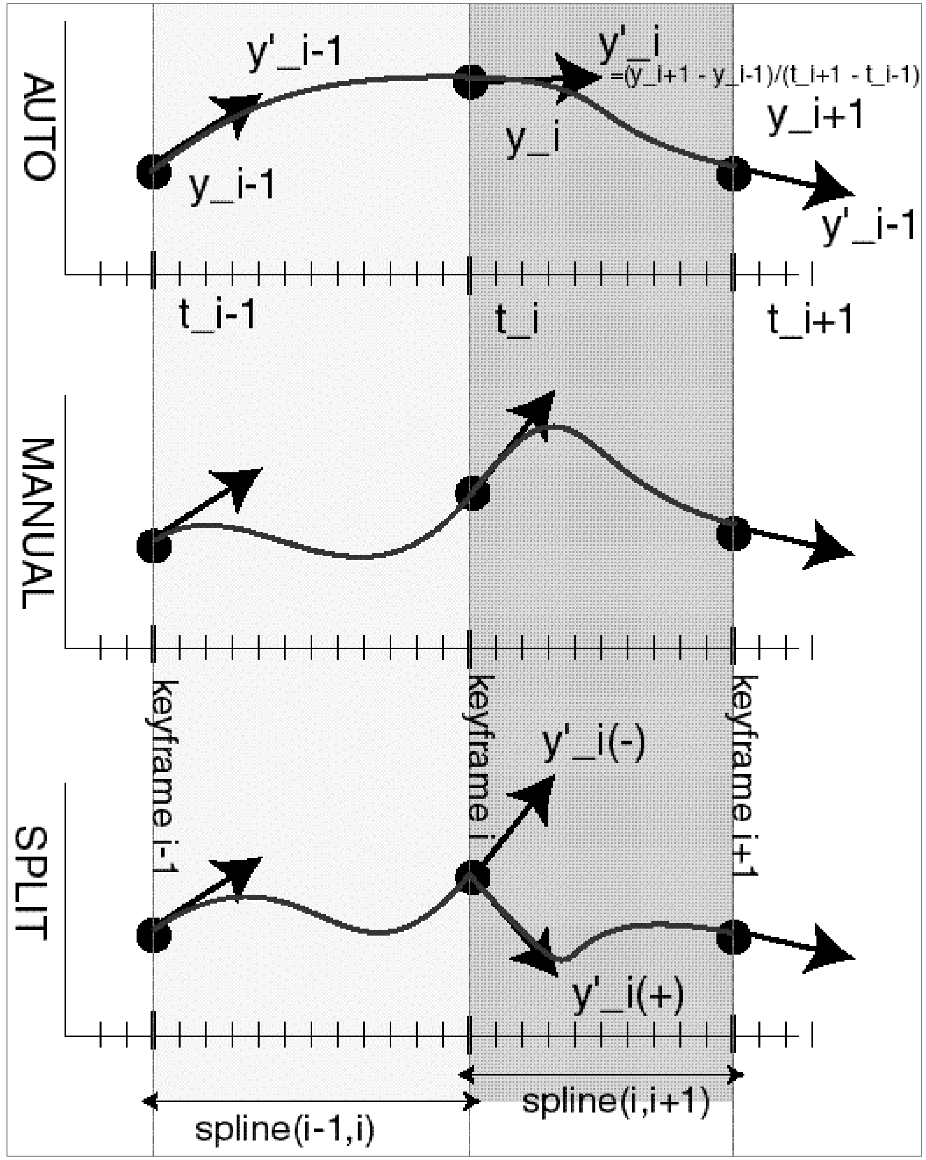


Splines for Interpolation

- Classic example - a ball bouncing under gravity
  - zero vertical velocity at start
  - high downward velocity just before impact
  - lower upward velocity after
  - motion produced by fitting a smooth spline looks unnatural
- What kind of continuity/control do we need?



How Do You Interpolate Between Keys?



Keyframing

- Specify the key frames
  - rigid transforms, forward kinematics, inverse kinematics
- Specify the type of interpolation
  - linear,cubic, parametric curves
- Specify the speed profile of the interpolation
  - constant velocity, ease-in,out, etc.
- Computer generates the in-between frames

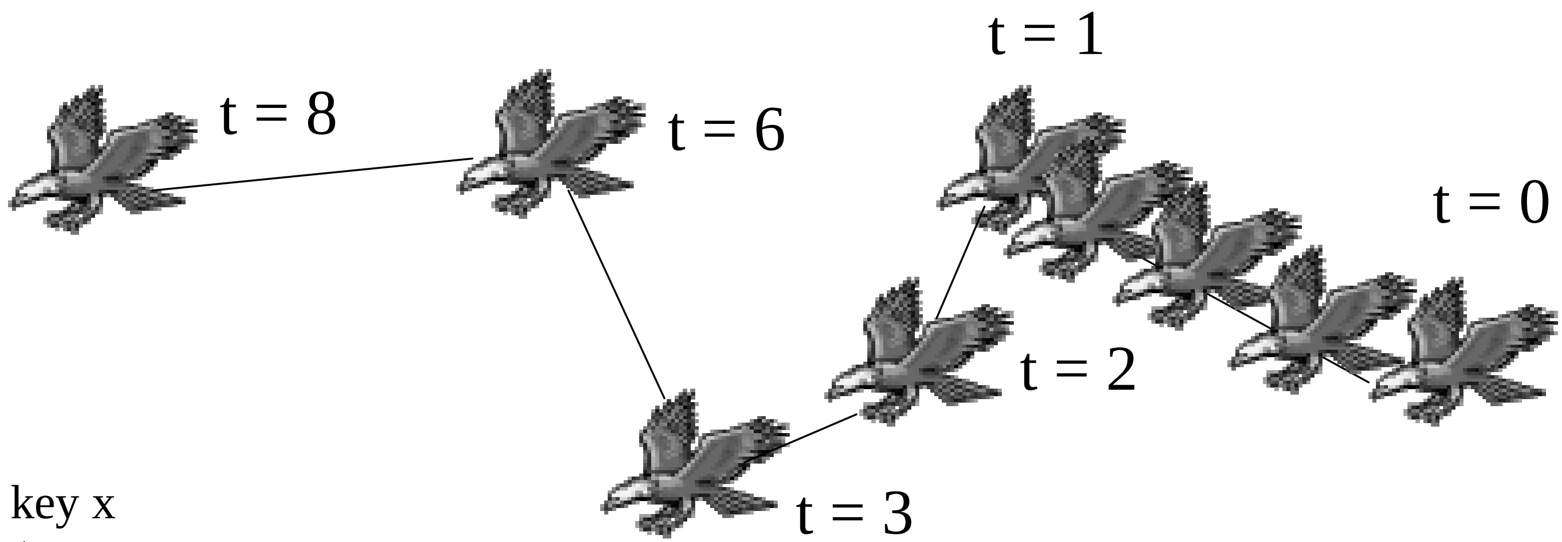
Keyframe Animation: Pros and Cons

- Gives good control over motion
- Eliminates much of the labor of traditional animation
  - But still very labor-intensive
- Impractical for complex scenes with everything moving: grass in the wind, water, and crowd scenes, for example

Now, in more detail:
how to interpolate and what to interpolate (positions for this lecture, orientations next time)

Linear Interpolation

Using linear "arcs" between keyframes



start key x

$$x = x_0 + \frac{t - t_0}{t_1 - t_0}(x_1 - x_0)$$

intermediate x

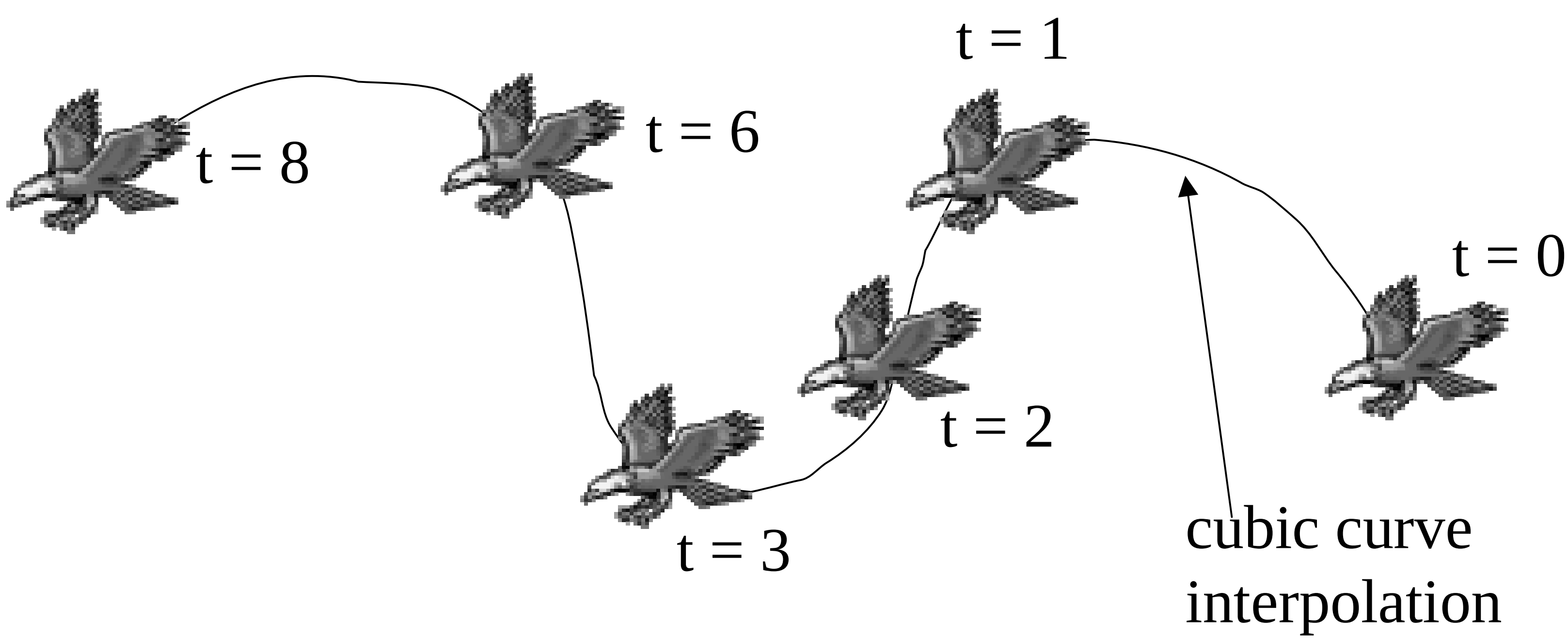
"a portion of"

total x distance between start and end key

where t_0 is the parameter value at the start of a segment, t_1 is the parameter value at the end of a segment and t is the parameter value for which you want to find the position, x

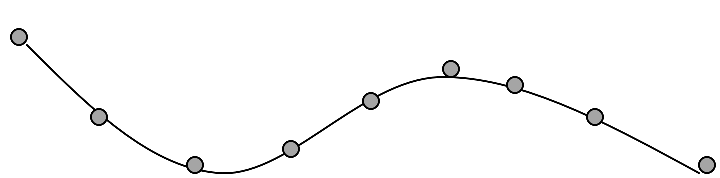
Cubic Curve Interpolation

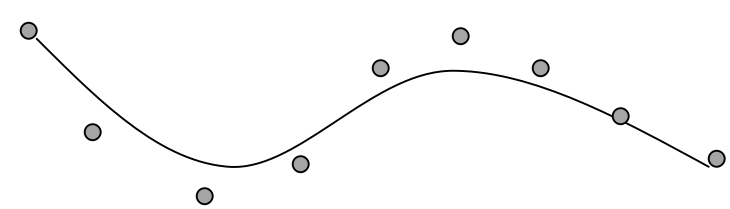
- Like a thin strip that can be bent to interpolate the points of interest



Cubic Curves

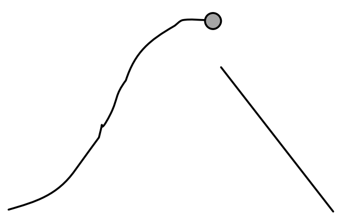
- Interpolating or Approximating?
  - –Interpolating curve passes through points
  - –approximating curve passes near the points used as weights or control points
  - –Hermite and Catmull-Rom are interpolating
  - –Bezier and B-spline are approximating
  - –Interpolating for data fitting, approximating ok for UI



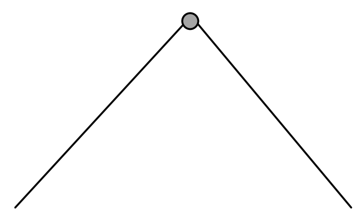


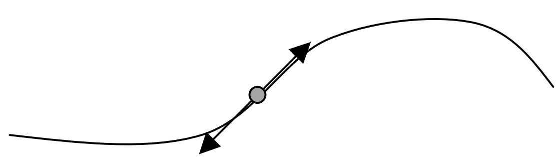
Continuity

Continuity between segments



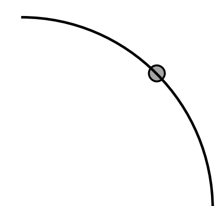
Positional (0^{th})





Tangential (1^{st})
Usually good enough for animation

Curvature(2^{nd})
Sometimes required
for modeling

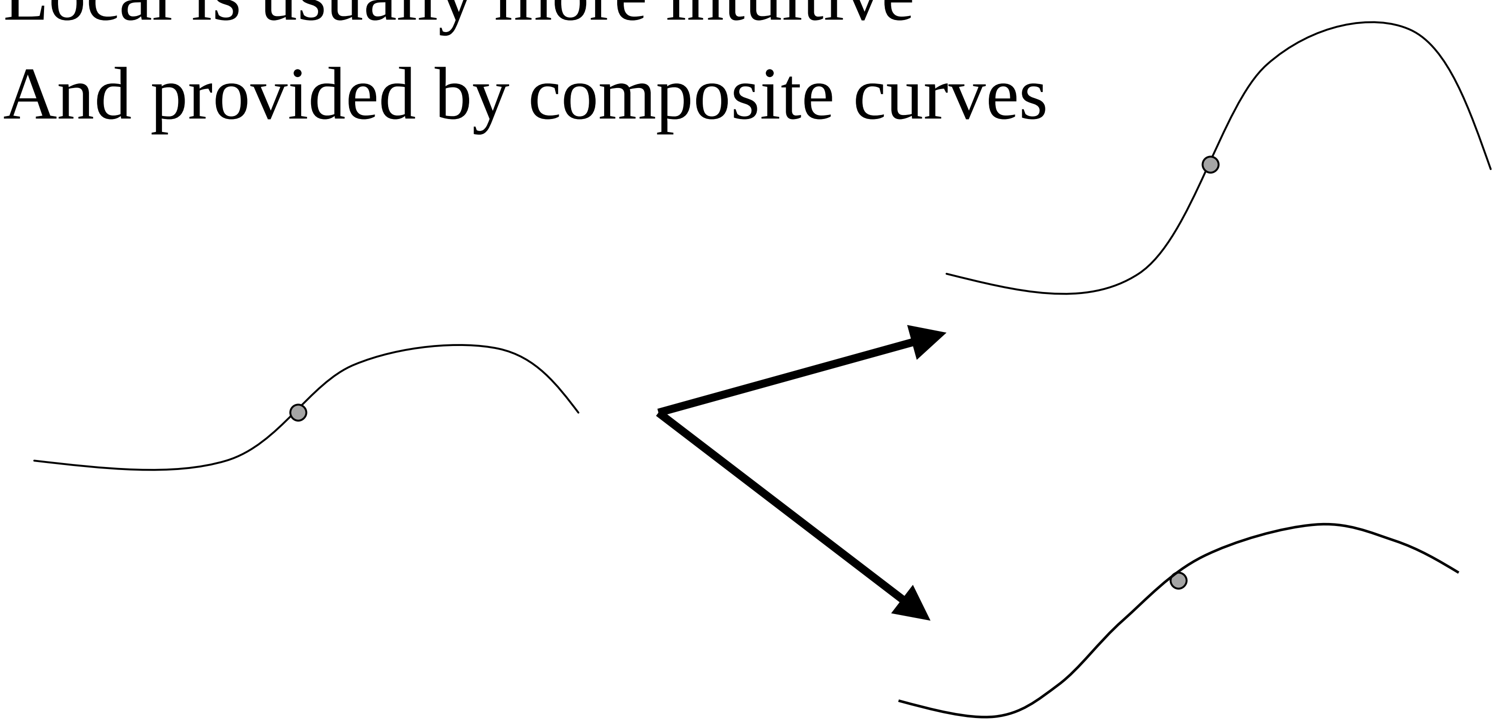


Global vs. Local Control

Does a small change change the whole curve or just a small segment?

Local is usually more intuitive

And provided by composite curves



Example Parametric Curve

- Hermite Curves – cubic polynomial

$$P(u) = au^3 + bu^2 + cu + d$$

- Really represents 3 equations in 3 space, one for x, y, and z
- Hermite interpolation requires
  - end points
  - tangent vectors at endpoints (tangent = derivative)
- To create a compound curve , use the end of one as the beginning of the other and share the tangent vector (first order continuity)

Hermite Curve Formulation

- Cubic Polynomial and its derivative

$$P_x(u) = a_x u^3 + b_x u^2 + c_x u + d_x$$

$$P'_x(u) = 3a_x u^2 + 2b_x u + c_x$$

- We need to solve for a,b,c,d for the hermite version
- We're given
  - $P_x(0)$, $P_x(1)$ - the endpoints
  - $P'_x(0)$, $P'_x(1)$ - the tangents

Hermite Curve Formulation

- Evaluate the two equations at the four given points
- Have 4 equations
  - P(0), P(1)
  - P'(0), P'(1)
- Have 4 unknowns a, b, c, d
- Solve

Hermite Curve Formulation

- Solution

$$a_x = 2(P_x(0) - P_x(1)) + P'_x(0) + P'_x(1)$$
$$b_x = 3(P_x(1) - P_x(0)) - 2P'_x(0) - P'_x(1)$$
$$c_x = P'_x(0)$$
$$d_x = P_x(0)$$

Parametric Curves in Matrix Form

$$P(u) = au^3 + bu^2 + cu + d$$

$$P(u) = U^T M B$$

where U is the variable..or parameter
M is the coefficient matrix
B is the geometric information

$$U^T = \lfloor u^3 u^2 u^1 1 \rfloor$$

$$P(u) = \lfloor u^3 u^2 u^1 1 \rfloor M B$$

$$M = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} P_i \\ P_{i+1} \\ P_{i'} \\ P_{i'+1} \end{bmatrix}$$

Splines

Should be familiar from 15-462 Computer Graphics

Review

- On web page: handout
- Appendix B.4 in Parent

Keyframing Recipe

- Specify the key frames
  - –rigid transforms, forward kinematics, inverse kinematics
- Specify the type of interpolation
  - –linear,cubic, etc. parametric curves
- Specify the speed profile of the interpolation
  - –constant velocity, ease-in,out, etc.
- Computer generates the in-between frames using this information

Now what?

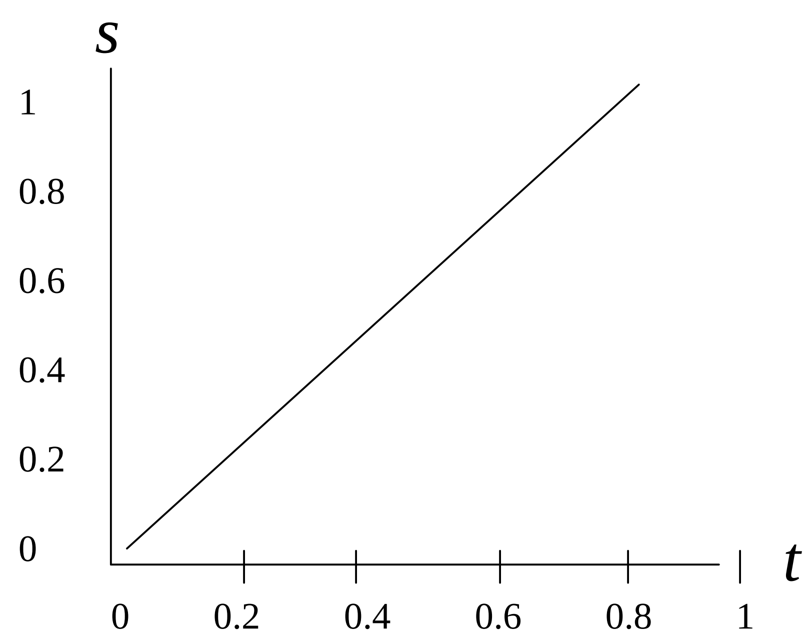
- We have key frames or points
- We have a way to find the in-between points
- Now we want to specify velocity of the interpolation...or, given a parameter value, we want to be able to say how far the object travels along the interpolation curve in a given period of time

Speed Curves (in Parent's terminology)

Speed Control

Maps parameter such as time to arclength

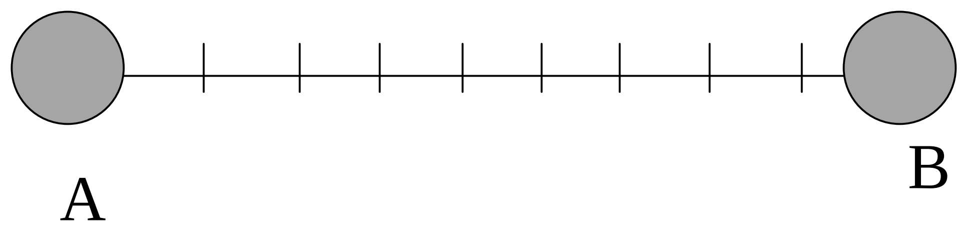
Simplest form is to ask for constant velocity along the path



where s is the arclength or distance traveled along the space curve

Speed Control

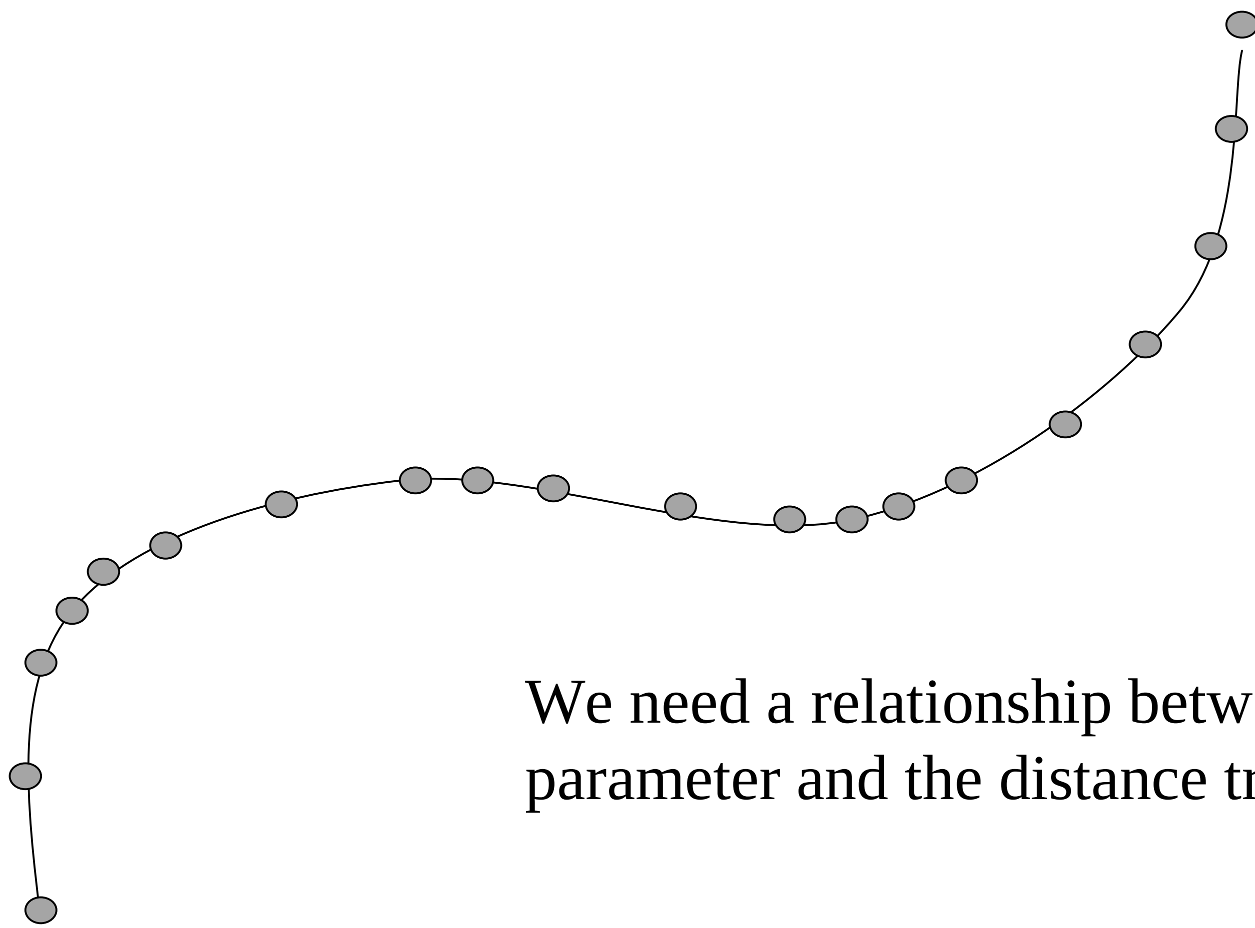
- Say we want a constant velocity interpolation from key A to key B (for each time increment, the position increases a constant amount)



- In the linear interpolation case, we can use time as the parameter, increment it a constant amount, and evaluate it to get the new position
- But, there’s a problem in the cubic polynomial case…

Non-Uniformity in the Parameter

- The problem is that uniform steps in the parameter don’t correspond to uniform path distances



We need a relationship between the parameter and the distance traveled

Arclength Reparameterization

- Reparameterize the space curve by arclength
- Problem
  - Given a parametric curve and 2 parameter values u1 and u2, find LENGTH(u1, u2)
  - Given an arc length S, and parameter value u1, find u2 such that LENGTH(u1,u2) = S.
- Not possible analytically for most curves (B-splines, for example)

Finite Differences

- Sample the curve at small intervals of the parameter and determine the distance between samples
- Use these distances to build a table of arclength for this particular curve

Index	Parametric Value	Arc Length
0	0.00	0.000
1	0.05	0.080
2	0.10	0.150
3	0.20	0.230
4	0.25	0.320
.		
.		
.		

Finite Differences

If we want to know S at a particular value of the parameter u (=0.73), spacing of the table is 0.05

– Find the entry in the table closest to this u and use it

$$i = (\text{int})\left(\frac{given\ parametric\ values}{dist\ between\ entries} + 0.5\right) \qquad i = (\text{int})\left(\frac{0.73}{0.05} + 0.5\right) = 15$$

– or take the u before and after it and interpolate linearly

$$i = (\text{int})\left(\frac{given\ parametric\ values}{dist\ between\ entries}\right)$$

$$L = ArcLength[i] + \frac{(GivenValue - Value[i])}{(Value[i+1] - Value[i])}(ArcLength[i+1] - ArcLength[i])$$

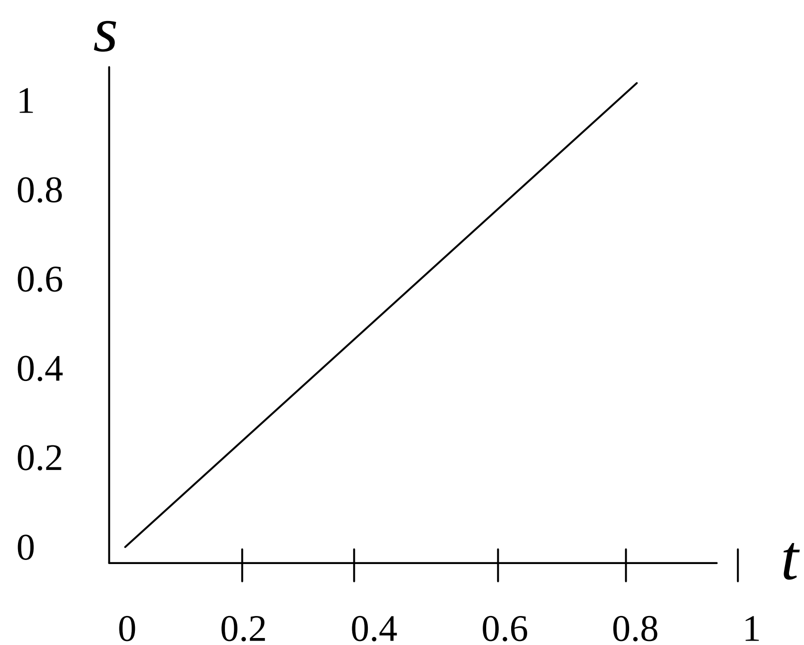
Finite Differences

- Use better interpolation for greater accuracy
- Use adaptive sub-sampling to handle curves with very different curvature in different portions of the curve

Similarly, we can find u given the desired arclength

Speed Control

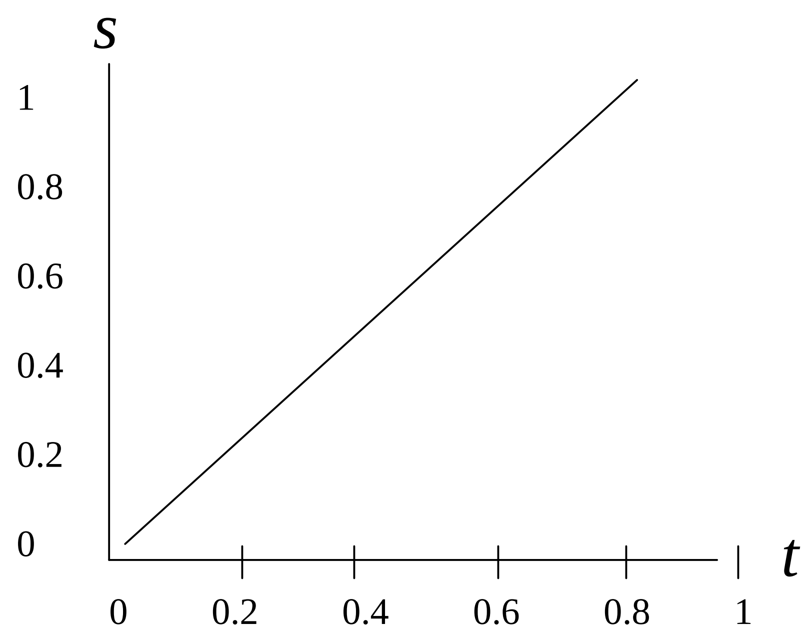
- Now we can control speed as follows
  - Given a time t, lookup the corresponding arclength s in the speed curve



Speed Control

– For the resulting arclength, s, look up the corresponding value of u in the reparameterization table
– Evaluate the curve at the resulting u to obtain the correct interpolated position for the animated object for the given time, t

Constant Velocity Speed Curve

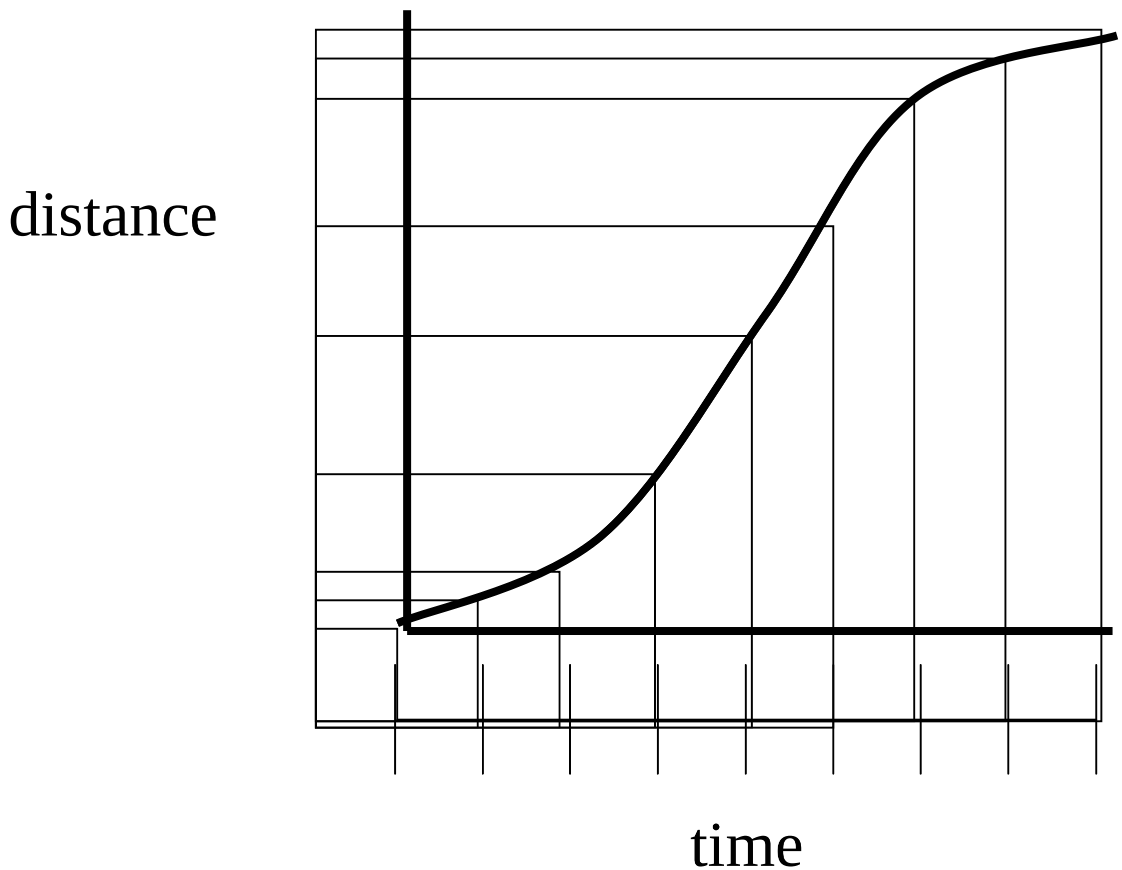


If our unit is meters, it’s moving at 1 m/s

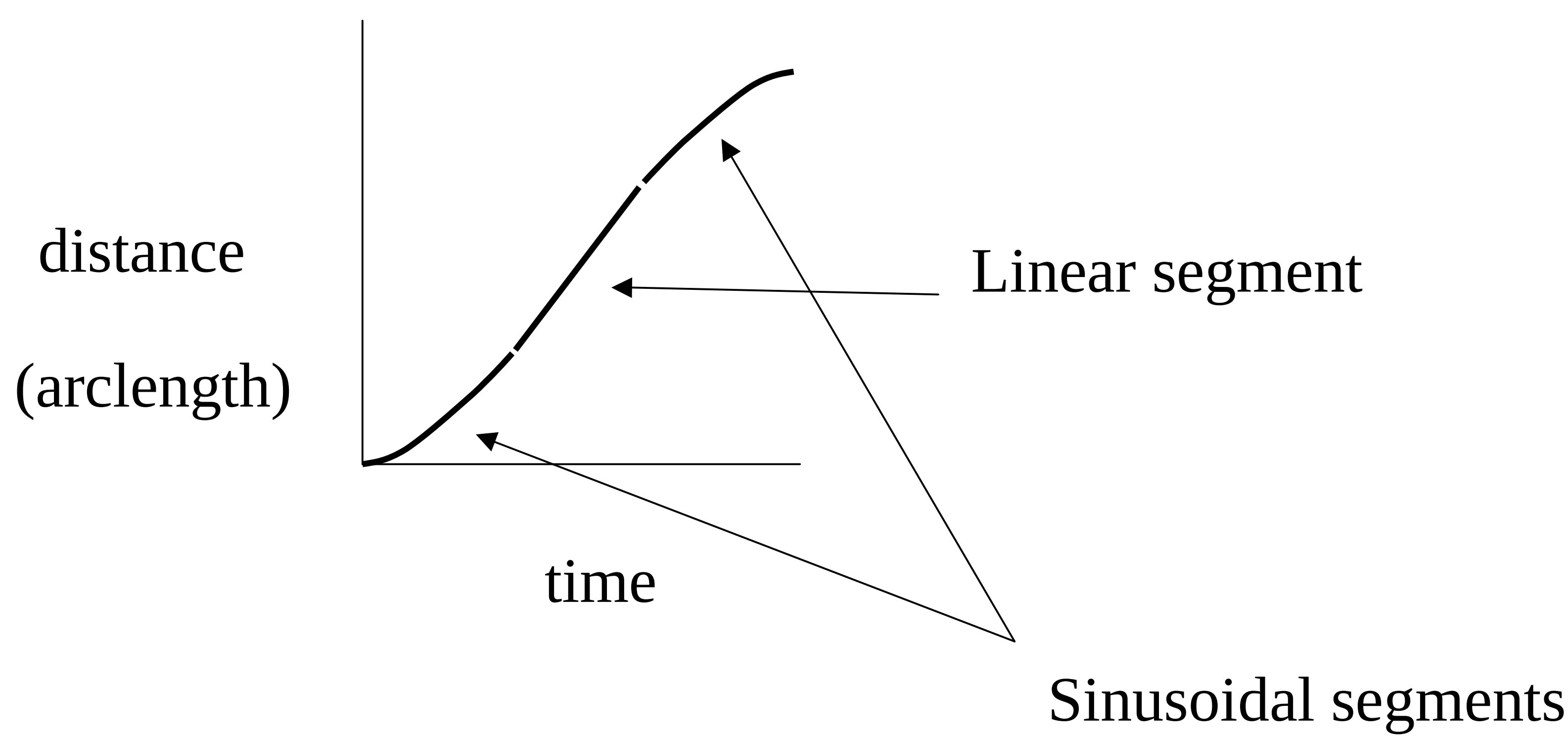
Too simple to be what we want…

Ease-in Ease-out curve

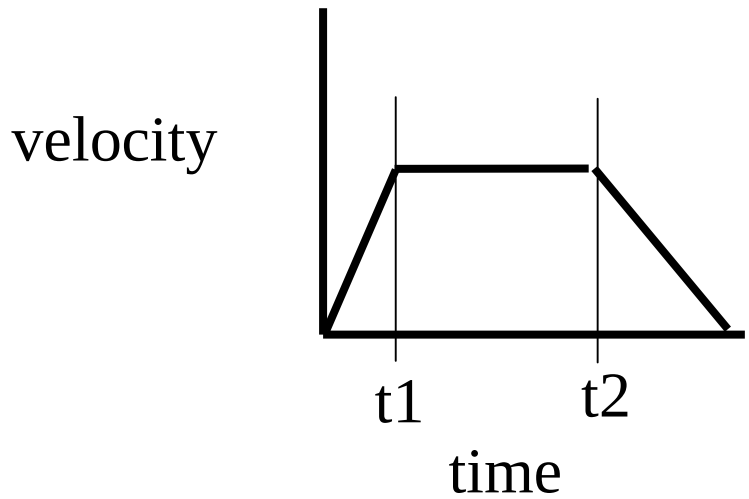
- Assume that the motion starts and stops at the beginning and end of the motion curve

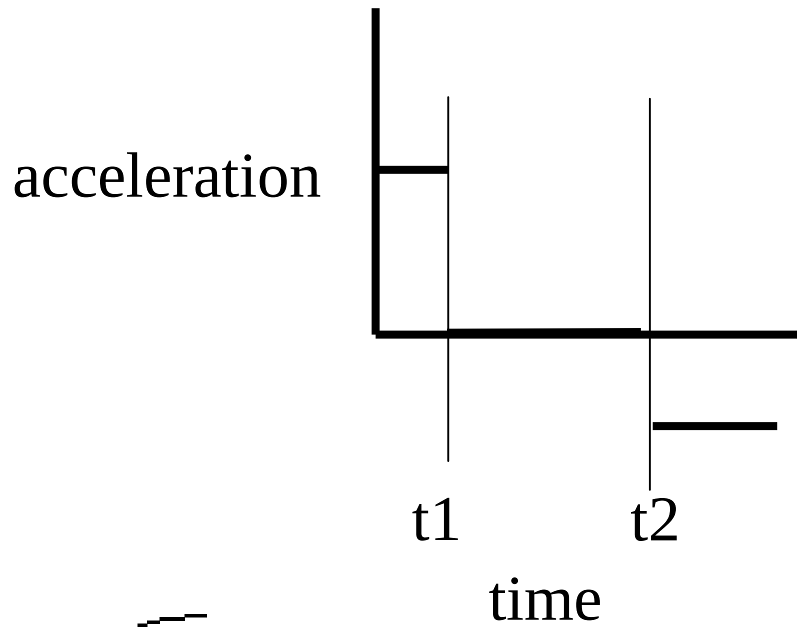


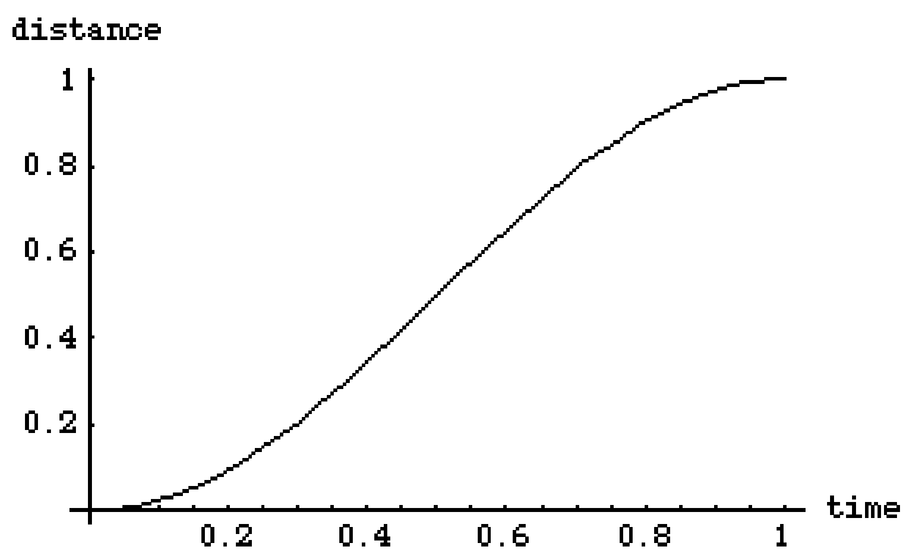
Piecing Together Curves for Ease In/Out



Integrating to avoid the sine function







Summary

- Space curve interpolates points – where to go!
- It’s parameterized by a variable that doesn’t produce equal increments in arclength
- reparameterize the arclength
- Speed curve defines how time relates to arclength – how far to go on each frame
- We design a speed curve to control arclength with time
- Use this arclength to determine u
- Use u to generate the in between positions…or color, intensity, etc

Keyframing: Issues

- What should the key values be?
- When should the key values occur?
- How can the key values be specified?
- How are the key values interpolated?
- What kinds of BAD THINGS can occur from interpolation?
  - Invalid configurations (pass through walls)
  - Unnatural motions
    - » Painful twists/bends
    - » Going the “long way around”
  - Jerky motion

Keyframe Animation: Production Issues

- How to learn the craft?
  - apprentice to an animator
  - practice, practice, practice
- Pixar starts with animators, teaches them computers and starts with computer folks and teaches them some art

What about rotations????

- Rotations are a special case
- We can interpolate rotations using these techniques….but there are some problems

We’ll talk about this on Thursday!