CogSci 109: Lecture 12

Tuesday Oct 31, 2006

Splines, error analysis, midterm practice

Announcements

- HAPPY HALLOWEEN!!!
- Midterm is Thursday!!!
- Review session location
- Be sure to refresh your browser's cache by pressing the reload button once the page has loaded <example>
- Decoupling what a variable represents from the variable name
- Example of sorting issues on homework

Quick review of material so far…

- <topics page>

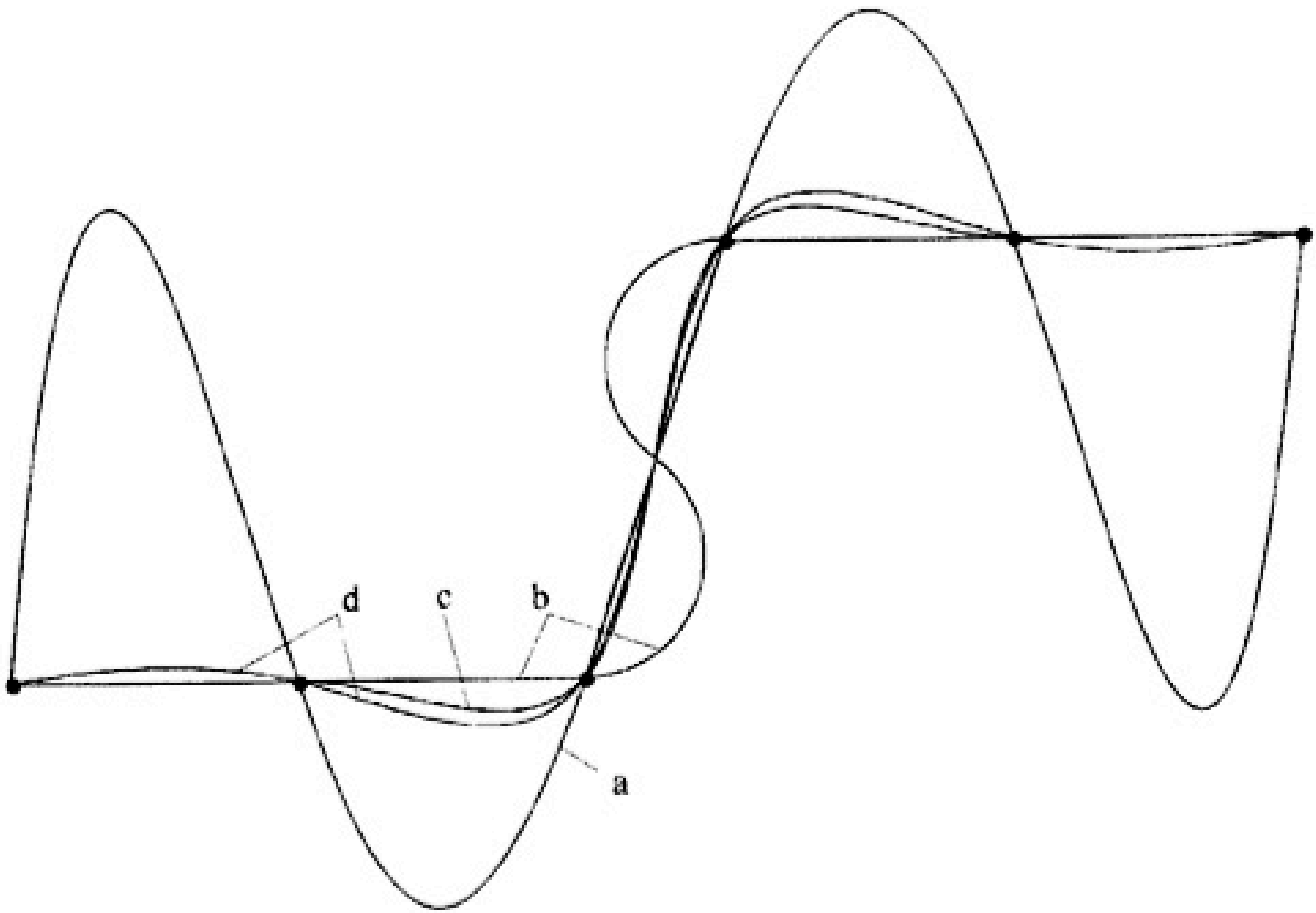
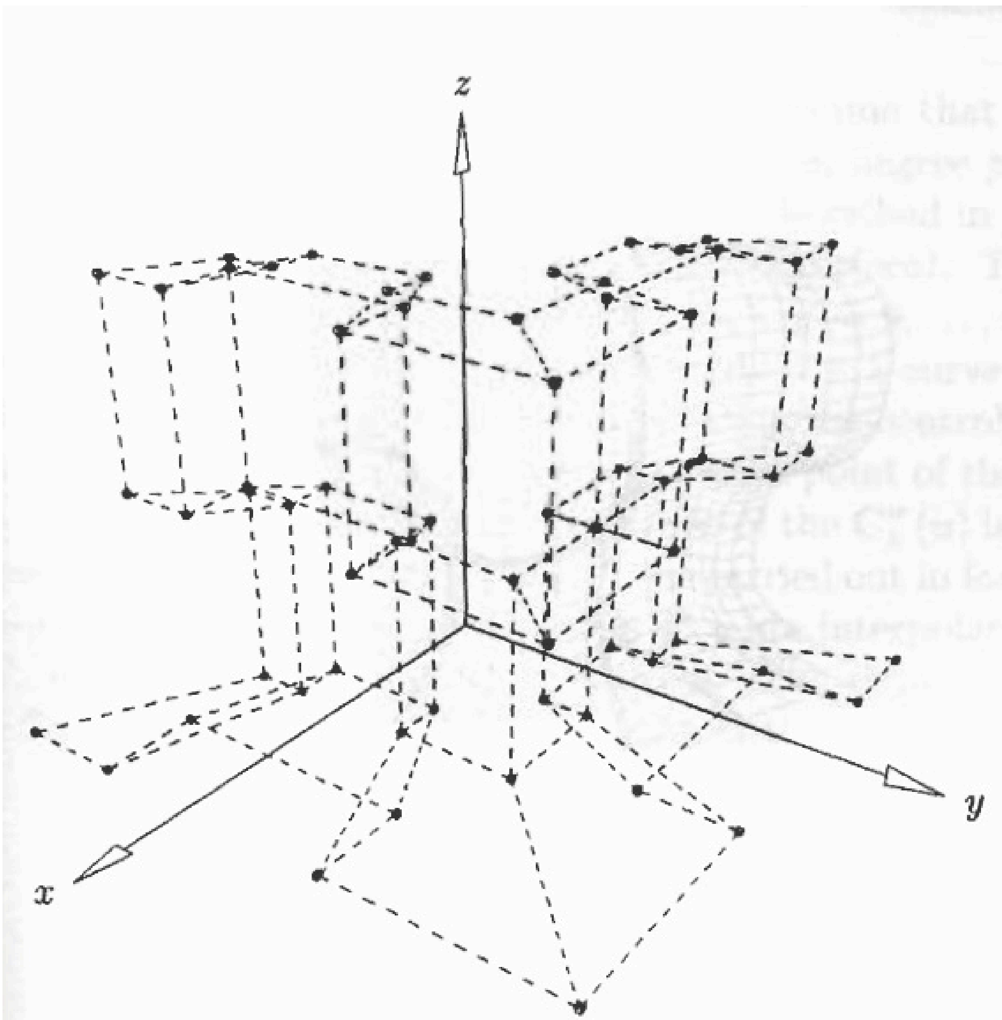


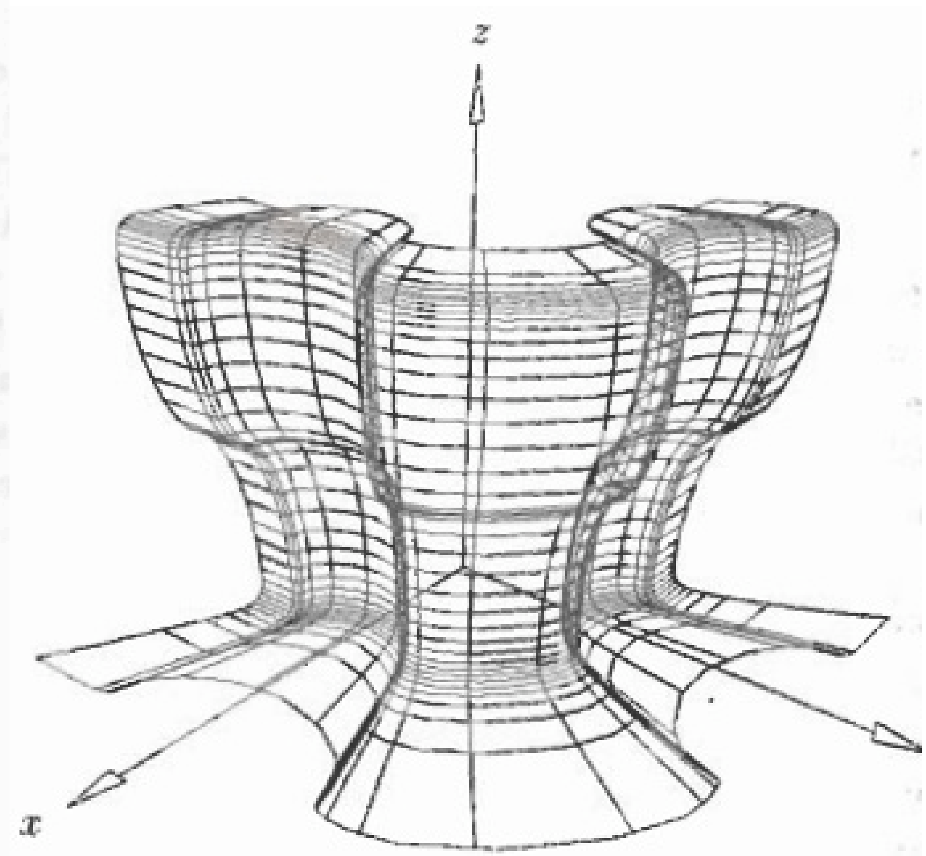
Figure 7.1 Interpolation curves drawn for six vertex points (dots), with y plotted in the vertical and x in the horizontal direction. Curves are shown for (a) a high-order polynomial fit, (b) a circular-arc fit, (c) a parabolic blend, and (d) a natural cubic spline.

Splines are useful in many places Lagrange fails

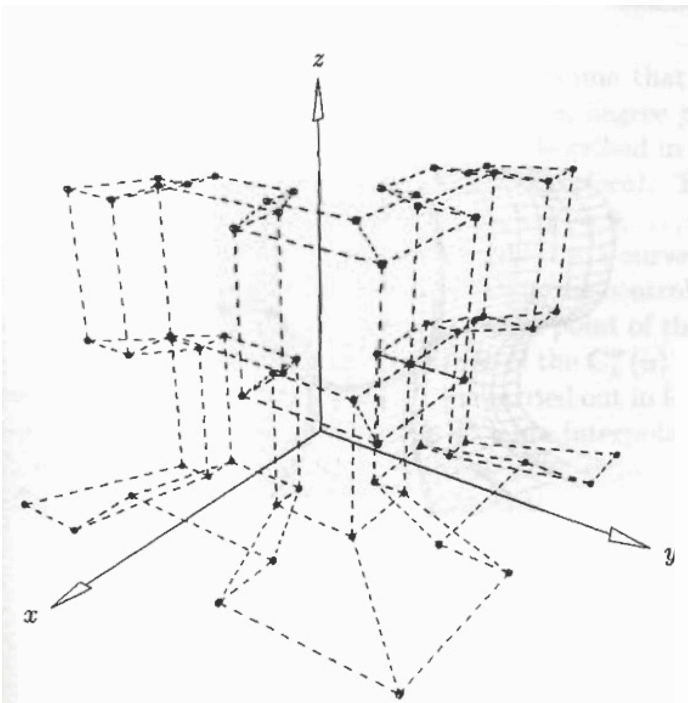
- Large number of data points
- Also can make a curve that passes through all data points
  - some types do not enforce this
- Drawn from drafting who drew from classical fine woodworking
  - Thin piece of wood stretched between pegs to create curves
  - Many types of splines dependent on end conditions
    - Pull tightly on the spline, curve gets sharper about the data points

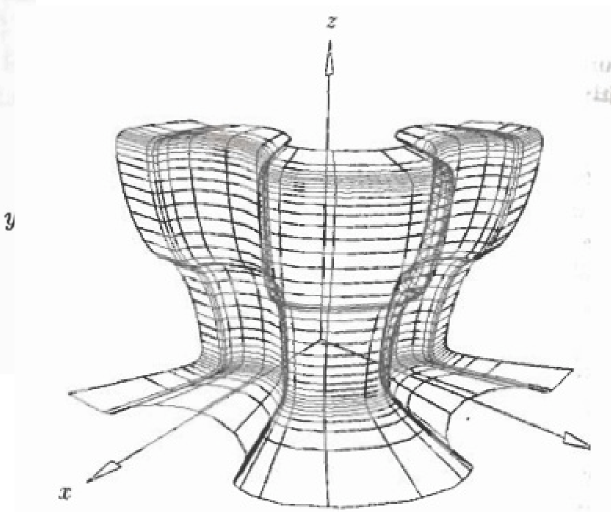
Splines are useful for N-Dimensions

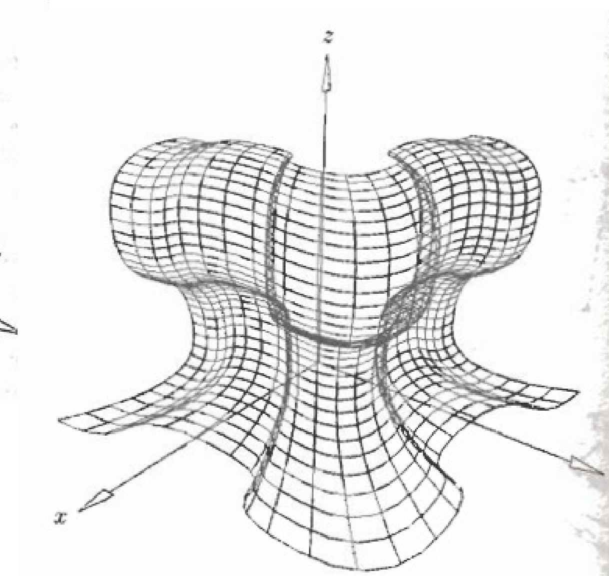




Splines also give you control over the final outcome of the curve





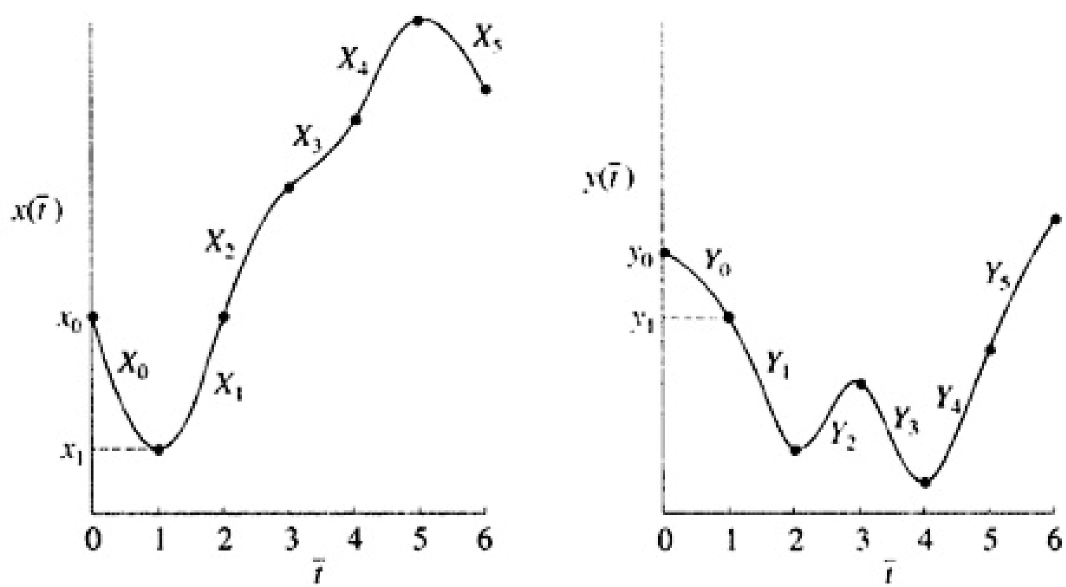


Some types of splines

- Natural cubic spline
- Quadratic B-Splines
- Hermite Cubic Splines
- Coons Cubic Splines
- Rational B-Splines
- NURBS (Non-Uniform Rational B-Splines)

Natural Cubic Spline - a conceptual introduction

- We construct the following curve in sections



Natural Cubic Splines

- We fit another parametric curve, with a value of t from 0-1 again and make the ith segment according to

$$Y_i(t) = a_i + b_i t + c_i t^2 + d_i t^3$$

- And we solve for each set of these constants by requiring continuity at the end points (one section smoothly flows into the next, and the slope must match as well)

$$Y_i(0) = y_i = a_i$$

$$Y_i(1) = y_{i+1} = a_i + b_i + c_i + d_i$$

$$Y_i'(0) = D_i = b_i$$

$$Y_i'(1) = D_{i+1} = b_i + 2c_i + 3d_i$$

Uncertainty

- Error does not mean, in science, mistake
  - It means the level of uncertainty in measurements and calculations
  - Can't eliminate by being careful, must instead minimize them
- Basically want to have an estimate which is as reliable as possible
  - 'keep an eye' on your uncertainty

Impossibility of certainty

- No physical quantity can be measured with absolute certainty
  - Wood door example
- Mathematical approximations to real systems are ALWAYS approximations, no matter how good
  - Any model you make is ONLY an approximation and should NEVER be confused with the real system
    - **Wrong**: "The Brain is computing the inverse of this matrix"
    - **Right**: "Our model approximates what the brain is doing by computing the inverse of this matrix"

The question…

- The question is not whether you are right or not
- The question is whether your approximation is good enough to be useful, dependent on what you consider to be ‘good enough’

Computing the estimated error

- One way to assess how good your model is consists of computing an estimated error
  - Typically you then decide whether your error is 'within bounds'
    - (you create a boundary, such as the error in measuring/predicting position of a limb in space must be less than 10 inches)
- Uses one of many possible methods

Different error estimates

- There are many ways to estimate errors, here are a couple of common ones
  - To get a single # - can use various norms
    - 2-norm

$$\|e\|_2 = \sqrt{\sum_i (y_i - \hat{y}_i)^2}$$

    - Mean-squared-error

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

  - Curve - simple error (for a time dependent signal y(t))

$$e(t) = y(t) - \hat{y}(t)$$

  - Curve - prediction error

$$e_p(t) = y(t) - \hat{y}(t \mid t-1)$$

All the mathematical equations we've done so far are different models!

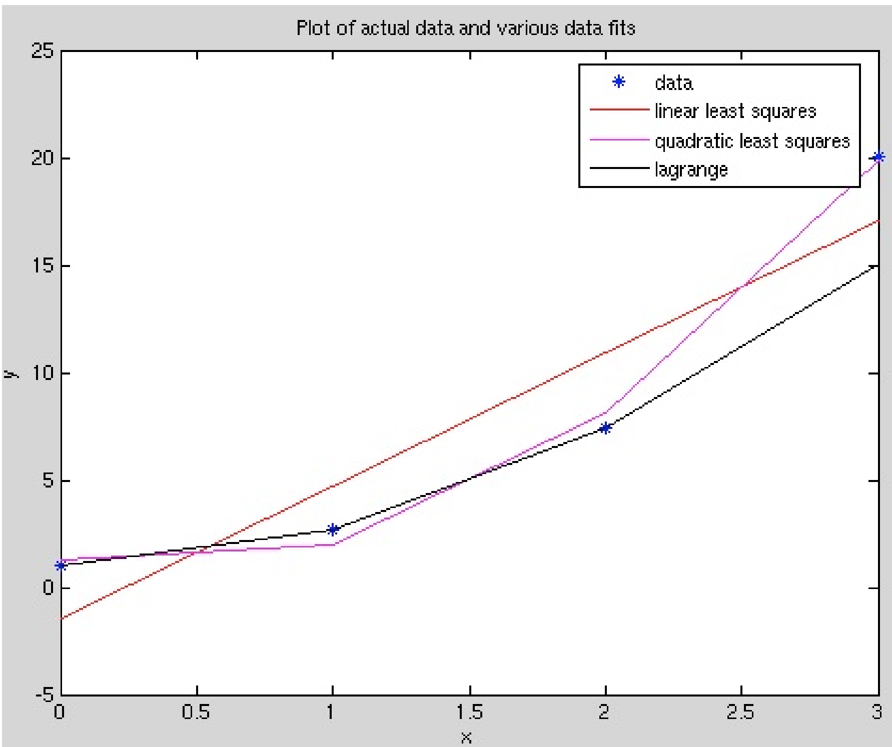
- Least squares is a very common method of fitting a model
- linear and nonlinear methods of interpolation are models of data approximation
  - Splines and lagrange

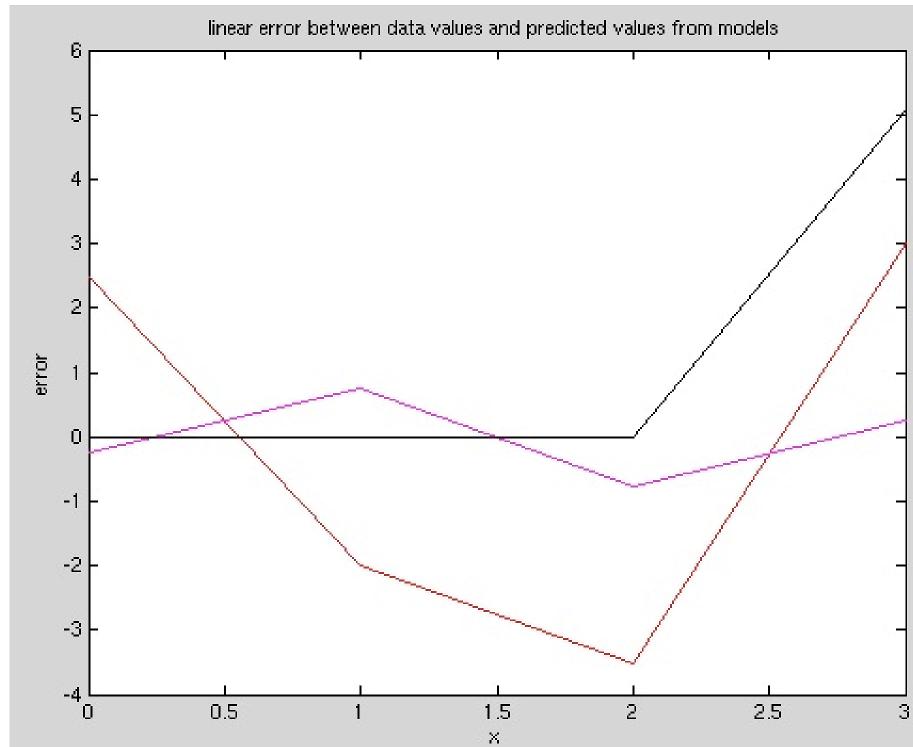
Different ways of modeling based on data

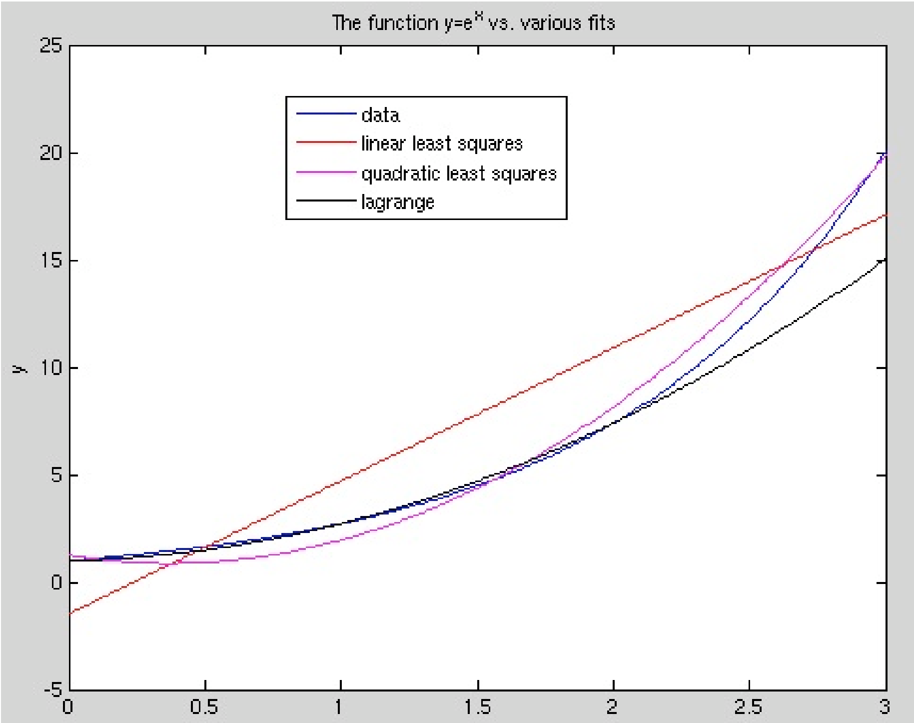
- Record all your data, then create a fit and study the resulting model
- Record all your data, split the recorded data into different groups
  - use one group to fit a model
  - then the other to check and see how well does your model predict what the system does (this is called model validation - or ‘invalidation’)

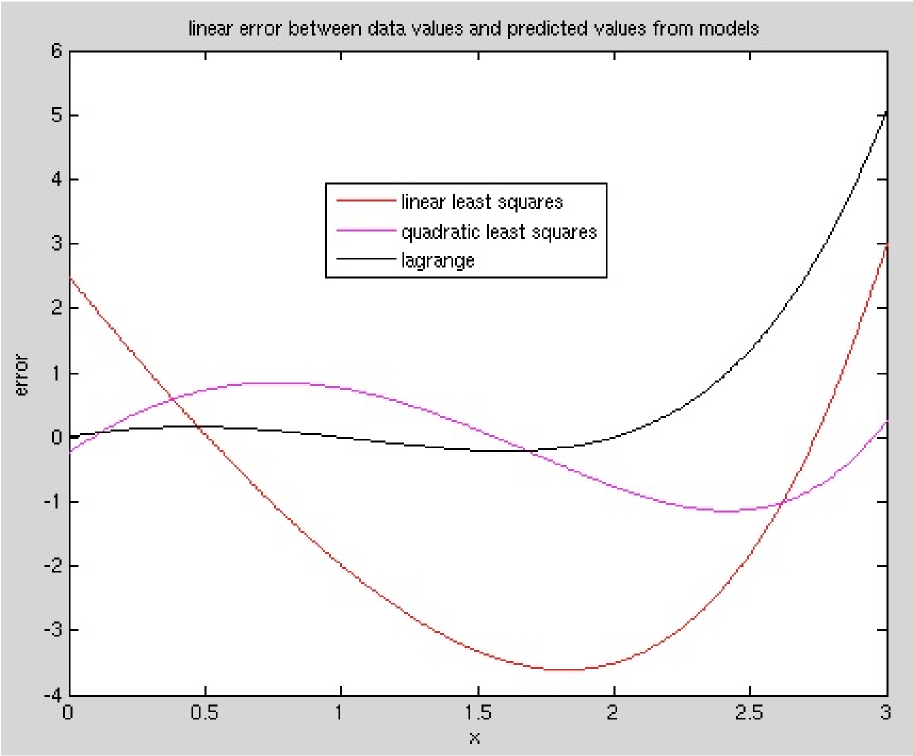
Example from the last few classes of the first method

- Approximation of $y = e^x$ using the various methods we know already
- I generated simulated data by computing y=exp(x) for a domain of [0, 1]at three points (0.0, 0.5, 1.0)
  - Then I created a linear least squares fit, a quadratic least squares fit, lagrange and cubic spline fits
  - I assess how well each model fit does by first plotting the error between the data and the different methods, then plotting the real function vs. the different methods along a continuous curve


Plot of actual data and various data fits
data
linear least squares
quadratic least squares
lagrange
y
x
25
20
15
10
5
0
-5
0
0.5
1
1.5
2
2.5
3


linear error between data values and predicted values from models
error
x
6
5
4
3
2
1
0
-1
-2
-3
-4
0
0.5
1
1.5
2
2.5
3


The function y=e^x vs. various fits
data
linear least squares
quadratic least squares
lagrange
y
25
20
15
10
5
0
-5
0
0.5
1
1.5
2
2.5
3


linear error between data values and predicted values from models
linear least squares
quadratic least squares
lagrange
error
x
6
5
4
3
2
1
0
-1
-2
-3
-4
0
0.5
1
1.5
2
2.5
3

We can also compute the error as a single quantity

- e2lls = 125.7192
- e2nlls = 10.9367
- e2lag = 86.3331
- From this we see that over this interval, the lagrange polynomial fits the data the best if we're trying to minimize this error as a criterion