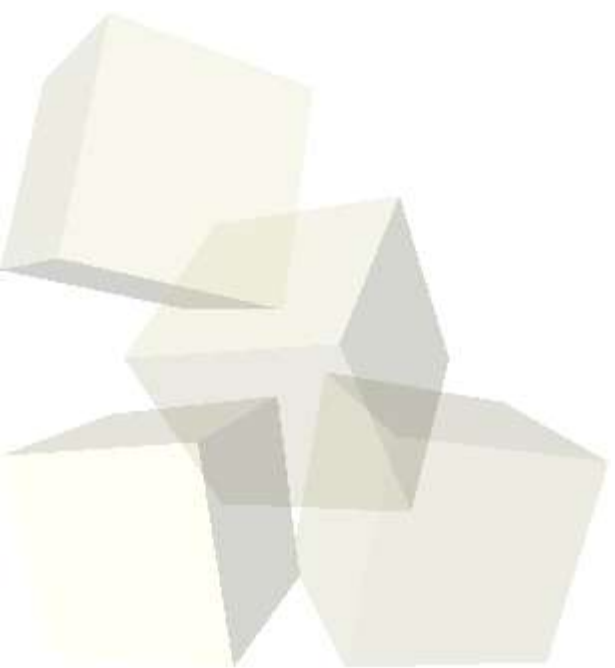




Newton's Method

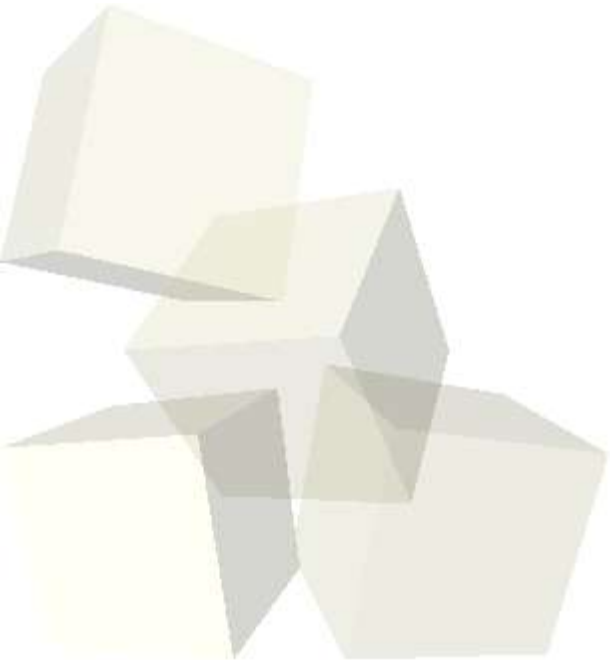
9-16-2005

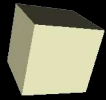




Opening Discussion

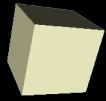
- What did we talk about last class?





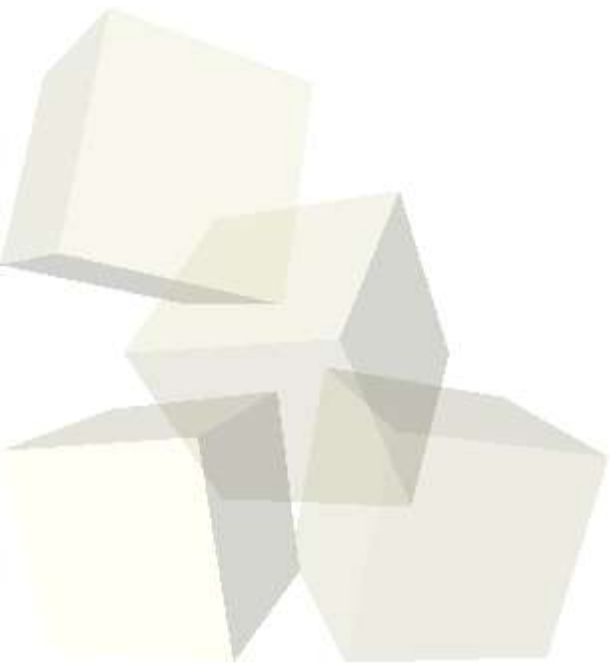
- Now we get into the things that Matlab was really developed for and where it really stands out.
- 2-D arrays are basically matrices and can be used for doing all types of math.
- Before we get into this we should talk about systems of linear equations and how they can be solved.
- A set of linear equations is typically expressed as $Ax=y$, where A is an n by n matrix and x and y are n by 1 matrices. You are given A and y and want to solve for x .
- If the equations are “well behaved” there is a single solution $x=A^{-1}y$. Use \backslash instead of $\text{inv}()$.

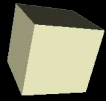
- If A isn't square there isn't a single solution. When A has more rows than columns there is no exact solution (overdetermined). When A has more columns than rows there are an infinite number of solutions (underdetermined).
- Matlab actually has routines that will try to solve systems of equations that aren't "well behaved". That is, it will approximate both over-determined and under-determined systems.
- In the overdetermined case, $A \setminus y$ will give you the least squares solution. This can be viewed as an optimal fit.



Sparse Matrices

- Matlab also has the ability to store sparse matrices.
- We aren't really going to take advantage of this in this class, but if you have a large matrix that has mostly zeros in it, this can be significant.





Root Finding

- A common mathematical problem is that of finding where a function has a certain value. Solving for the zeros of a function is called root finding, but since you can easily subtract any value it is very general.
- There are many methods to find roots. One of the most widely used is Newton's method. I get the feeling that Matlab uses a binary search method though.

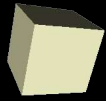




Newton's Method

- Newton's method uses a linear approximation to a function and follows the line to zero, then uses that as a better guess.
- $f(x) \sim f'(v)(x-v) + f(v)$
- If you don't know the analytic derivative you can use a secant method or take a numeric derivative.
- This method is very fast if you have a guess close to the root. It can be poorly behaved in some places though.

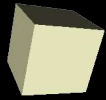




Binary Searching

- A binary search root finding method needs a range such that $f(x_1) \cdot f(x_2) < 0$ (the two endpoints must have a different sign).
- You can either cut the range in half or use a line approximation between the two endpoints.
- The trick in this method is finding the endpoints for the range. If a function has a very small region of negative values, for example, this can be hard to do.





- Another common problem is trying to find the smallest or largest values that a function takes in a certain range.
- Mathematically, these “local extrema” are points where the first derivative is zero so you just do root finding on that.
- “Hill climbing” methods come in many forms. Their simplest forms wind up working in a manner very similar to root finding.
- Matlab provides you will methods for doing minimization of functions.



- Assignment #3 is due on Monday.
- Quiz #2 is on Wednesday.

