CSCE/Math 4/841

So what is it? It is a branch of mathematics that attempts to answer questions like: How do we approximate a given type of function or dataset by another (somehow “better”) type of function? E.g., can we reasonably approximate the function

$$f(x) = \cos(x) \, , \, 0 \leq x \leq \pi/2$$

with a quadratic polynomial? A cubic? What if we are given data points through which the approximating function should run (as in computer graphics)? Can we run a smooth curve through or near the points? Conversely, can we approximate a signal given by some dataset by a trigonometric polynomial, so as to recover its modes? How do we even determine that an approximation is “close”?

Sources: textbook (“Approximation theory and methods” by M. J. D. Powell) and notes. In particular, I will provide the training for the official programming language of the course, Matlab, which we will need because this is a “hands-on” course. No prior exposure to Matlab is assumed. Formal prerequisites for this course are Math 221, 314 and a programming language. As a matter of fact, Math 221 and the programming language are optional, but Math 314 or an equivalent linear algebra course is essential.

Topics will include:

- Definitions of “best” along with existence and uniqueness of approximations.
- Polynomial interpolation
- Least squares approximations
- Orthogonal polynomials
- Trigonometric approximations and the FFT

- Splines and Bezier curves

We will be interested in the construction, accuracy and complexity of algorithms. Approximation theory is important to many areas of computation, e.g., computer graphics, signal processing and mathematical programming. Here is an example of an approximation problem that was posed to me by a professional game programmer: Supply fast algorithms for approximations to the trig and inverse trig functions that do not use floating point arithmetic and division (typical platform might be for doing fixed point arithmetic on an ARM cpu with no floating point unit). Now suppose that division is allowed, though it is a fraction as fast as the other arithmetic operations. What's the best you can do? Approximation theory can help.

For more information, contact the instructor:

Thomas Shores

Ph: 472-7233

Email: tshores1@math.unl.edu

Website: http://www.math.unl.edu/~tshores1/m441F09.html