MA 1600 Course Summary

Benjamin Ong
ongbw@mtu.edu

Department of Mathematical Sciences
Michigan Technological University

April 22, 2015

Michigan Tech
Create the Future

Course Aims

From the syllabus:

- Understanding approximations and their implications.
- Appreciating sequences and their implications.
- Experience using mathematics & computers to gain insight into real-life problems.

Michigan Tech
Create the Future

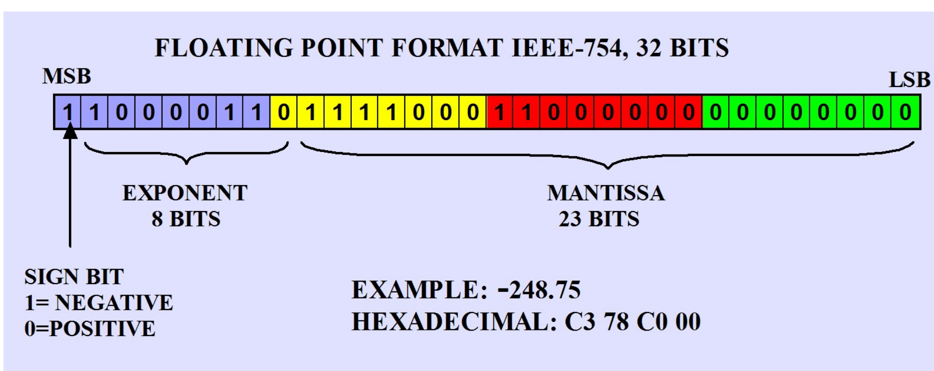
Understanding Approximations . . .

- Accuracy: how close is your approximation to the true solution? Precision: how many digits of your approximation do you trust?
- To quantify error:

$$error = |true\_solution - approximation|$$

  Many sources of error: measurement error, model error, truncation error, round-off error, discretization error, human error . . .
- round-off error: occurs because of finite-precision
  - Computer stores numbers in memory cells called bits
  - loss of precision: subtracting too almost equal numbers
  - amplification of error: dividing by a small number

Understanding Approximations . . .



- To quantify precision: significant digits / scientific notation.

$$3.1415 \times 10^0$$

Understanding Approximations . . .

Approximating data using polynomials.

- polynomial interpolation is useful, because polynomials are easy to evaluate, integrate and differentiate.

$$p(x) = \sum_{i=0}^{n} f_i L_{n,i}(x)$$

where

$$L_{n,i}(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{(x - x_j)}{(x_i - x_j)}$$

$$= \frac{(x - x_0)}{(x_i - x_0)} \frac{(x - x_1)}{(x_i - x_1)} \cdots \frac{(x - x_{i-1})}{(x_i - x_{i-1})} \frac{(x - x_{i+1})}{(x_i - x_{i+1})} \cdots \frac{(x - x_n)}{(x_i - x_n)}$$

Understanding Approximations . . .

Once you have the polynomial interpolant

- can evaluate polynomial at x^* to compute an interpolated value
- can take derivative of polynomial, and evaluate $p'(x^*)$ to approximate rate of change of data. Leads to difference formulas.
- can integrate polynomial, $\int_a^b p(x)$ to approximate integral of data on $[a, b]$. Leads to quadrature rules

Understanding Sequences and their implications . . .

When we compute approximate solutions, we are interested in

- how accurate / precise is our approximation
- if we are using an iterative algorithm, how quickly are we converging to an accurate/precise solution

Let $x_1, x_2, x_3, \ldots$ be a sequence obtained from an iterative algorithm. We say that $\{x_k\}$ converges to the solution x if

$$\lim_{k\to\infty} x_k = x$$

or equivalently

$$\lim_{k\to\infty} e_k = \lim_{k\to\infty} |x_k - x| = 0$$

Michigan Tech
Create the Future

Understanding Sequences and their implications . . .

- bisection algorithm:

  $$error = \frac{\text{size of interval bracketing root}}{2}$$

  Each iteration, error goes down by a factor of 2: linear convergence.
- finite difference approximations:

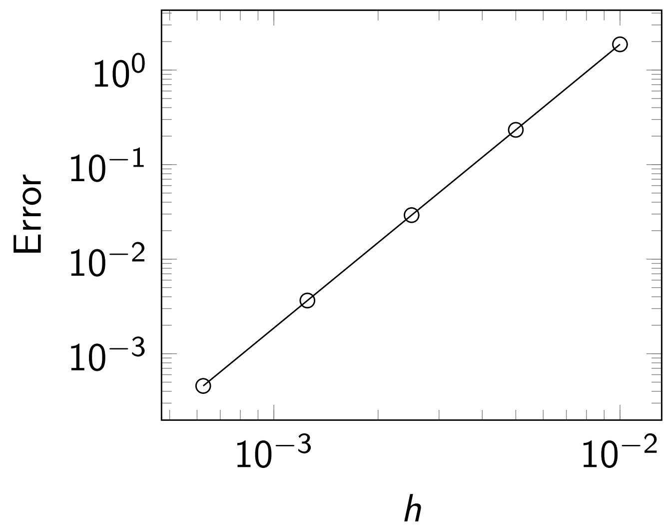
  $$\left.\frac{dy}{dx}\right|_{x_i} \approx \frac{y(x_i + h) - y(x_i - h)}{2h}$$

  second order convergence:

  $$\lim_{h\to 0} e_h = \mathcal{O}(h^2)$$

Michigan Tech
Create the Future

Understanding Sequences and their implications . . .



Rate of convergence computed by finding slope of best fit line to *loglog* data. In this example, rate of convergence $= 3$.

Real-Life Problems I

- Mortgage payments – root finding
  - bisection method
    - requires bracketing a root $(f(a)f(b) < 0)$
    - always converges to one of the roots in the defined bracket
    - slow convergence
  - modified bisection
  - newton's method:
    - uses information about the derivative (slope)
    - might diverge if function is non-smooth
    - if it converges, converges at second order
    - hard to find starting iterate which leads to convergence
  - secant method:
    - finite difference approximation to the derivative
    - might diverge if function is non-smooth
    - if it converges, converges at $\frac{1+\sqrt{5}}{2}$
    - hard to find starting iterate which leads to convergence

Real-Life Problems II

- Modeling Apple stock – interpolation / extrapolation / least squares
  - split data into training / validation set.
  - extrapolation is unstable
  - high-order polynomial interpolation is oscillatory
  - least squares to "best-fit" data
- game of life – deterministic systems
  - Final system state only dependent on initial condition
  - Deterministic systems: no randomness involved in the future state of the system.
  - from random initial conditions, formation of structures, patterns

Michigan Tech
Create the Future

Real-Life Problems III

- Random Walks (Brownian Motion)
  - estimating probability and expectation
  - random number generation
  - slow convergence. turns out: convergence looks like $\frac{1}{\sqrt{N}}$, where N is the number of experiments.
  - First example of a Markov chain: the state X_n depends only on X_{n-1}.
- Agent Based models
  - Robots in 2D.
  - Also a Markov chain system.
  - increasing intelligence
  - effect of noise and fuzzy logic

Michigan Tech
Create the Future

Real-Life Problems IV

- Solving Ordinary Differential Equations.
  - consists of a differential equation
  - initial condition
- Types of ODEs modeled:
  - Gravity – with and without drag
  - Population Growth/Decay
  - Harmonic Oscillators
  - Volterra equations – predator-prey models
- We worked on:
  - forward Euler (first-order)
  - improved Euler (second-order)
  - Runge-Kutta. (RK4: fourth order)
- built-in MATLAB routines: `ode45`

Michigan Tech
Create the Future

Real-Life Problems V

- Heat Equation – Solving PDEs
  - Partial Differential Equations consist of:
    - equation that describes physics of interest
    - Domain of interest
    - Initial Conditions
    - Boundary Conditions
  - apply (centered) finite difference approximations (in space) to get a system of ODE's
  - use forward Euler to solve ODE's.
  - observed first order convergence in time, second order in space
- Continuous Optimization – Steepest Descent Algorithms
  - iterative method
  - Move in direction of negative gradient (slope)
  - if stepsize is too large, shrink stepsize iteratively until small enough

Michigan Tech
Create the Future