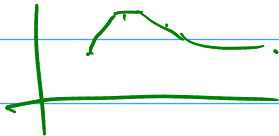


Overview

- Interpolation



▷ Error

▷ Application

- Monte Carlo

Exam overlooking
RELATE
maintenance
recruiting

HW2

Vandermonde Linear Systems

Polynomial interpolation is a critical component in many numerical models.

$$(x_1, y_1) \dots (x_n, y_n)$$

$$p(x) = a_0 \underline{1} + a_1 \underline{x} + a_2 \underline{x^2} + \dots + a_{n-1} \underline{x^{n-1}}$$

$$p(x_i) = y_i$$

points ↓

$$\begin{pmatrix} 1 & x_1 & x_1^{n-1} \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

↑ functions

$n \times n$

Demo: Polynomial Approximation with Point Values

Error in Interpolation

How did the interpolation error behave in the demo?

To fix notation: f is the function we're interpolating. \tilde{f} is the interpolant that obeys $\tilde{f}(x_i) = f(x_i)$ for $x_i = x_1 < \dots < x_n$.
 $h = x_n - x_1$ is the interval length.

$$O(h^n)$$

What is the error at the interpolation nodes?

$$0$$

Care to make an unfounded prediction? What will you call it?

convergence of order n
 n -th order convergence

Proof Intuition for Interpolation Error Bound

Let us consider an interpolant \tilde{f} based on $n = 2$ points so

$$\tilde{f}(x_1) = f(x_1) \quad \text{and} \quad \tilde{f}(x_2) = f(x_2).$$

Prove the interpolation error bound in this case.

$$E(x) = f(x) - p(x)$$

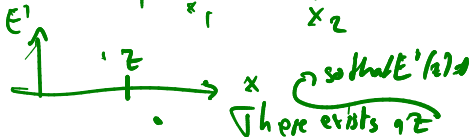
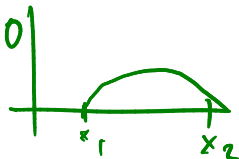
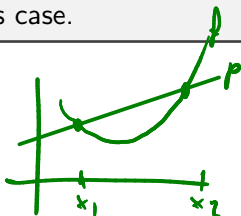
$$E(x_1) = 0 \quad E(x_2) = 0$$

$$\int_{x_1}^{x_2} E'(x) dx = E(x_2) - E(x_1) = 0$$

What does that say about E' ?

Case I: $E'(z) = 0$

Case II: It has a non-zero value



$$E(x) = \overbrace{E(x_1)}^0 + \int_{x_1}^x E'(w_0) dw_0$$

$$= \int_{x_1}^x \cancel{E'(z)} + \int_z^{w_0} E''(w_1) dw_1 dw_0$$

$$= \int_{x_1}^x \int_z^{w_0} E''(w_1) dw_1 dw_0$$

$$\left[E(x) = f(x) - \frac{p(x)}{ax+b} \right] E''(x) = f''(x)$$

$$C = \max_{x \in (x_1, x_2)} |f''(x)|$$

$$\leq \int_{x_1}^x \int_z^{w_0} C dw_1 dw_2 \leftarrow$$

$$\leq C \cdot \frac{h^2}{2} = \left(\max |f''(x)| \cdot \frac{h^2}{2} \right)$$

Making Use of Interpolants

Suppose we can approximate a function as a polynomial:

$$f(x) \approx a_0 + a_1x + a_2x^2 + a_3x^3.$$

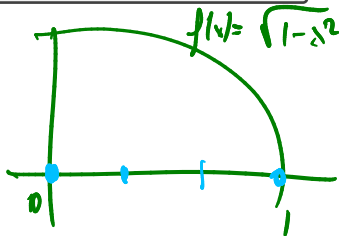
How is that useful? E.g. what if we want the integral of f ?

Step 1: Interpolate,
get polynomial
 $p(x)$.

Step 2: Claim that

$$\int_0^1 f(x) dx \approx \int_0^1 p(x) dx$$

Step 3: 4 · result $\approx \pi$



Demo: Computing π with Interpolation

$$p(x) = \underbrace{} \cdot \underbrace{1}_{\substack{\uparrow \\ \varphi_0(x)=1}} + \underbrace{} \cdot \underbrace{x}_{\substack{\uparrow \\ \varphi_1(x)=x}} + \underbrace{} \cdot \underbrace{x^2}_{\substack{\uparrow \\ \varphi_2(x)=x^2}}$$

$$\varphi_i(x) = x^i$$

$$\varphi_i(x) = \sin(ix)$$

$$\varphi_0(x) = \text{[graph of a pulse function]}$$

$$\varphi_1(x) = \text{[graph of a pulse function]}$$

More General Functions

Is this technique limited to the **monomials** $\{1, x, x^2, x^3, \dots\}$?

$$V = \begin{pmatrix} \cancel{1} & \cancel{x} & \cancel{x^2} & \dots & \dots & \dots \\ \varphi_0(x_1) & \dots & \dots & \dots & \dots & \varphi_{n-1}(x_1) \\ \vdots & & & & & \\ \varphi_0(x_n) & \dots & \dots & \dots & \dots & \varphi_{n-1}(x_n) \end{pmatrix}$$

Interpolation with General Sets of Functions

For a general set of functions $\{\varphi_1, \dots, \varphi_n\}$, solve the linear system with the generalized Vandermonde matrix for the coefficients (a_1, \dots, a_n) :

$$\underbrace{\begin{pmatrix} \varphi_1(x_1) & \varphi_2(x_1) & \cdots & \varphi_n(x_1) \\ \varphi_1(x_2) & \varphi_2(x_2) & \cdots & \varphi_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(x_n) & \varphi_2(x_n) & \cdots & \varphi_n(x_n) \end{pmatrix}}_V \underbrace{\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}}_a = \underbrace{\begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{pmatrix}}_f.$$

Given those coefficients, what is the interpolant \tilde{f} satisfying $\tilde{f}(x_i) = f(x_i)$?

$$\tilde{f}(x) = \underline{a_1} \cdot \varphi_1(x) + \underline{a_2} \cdot \varphi_2(x) + \dots + \underline{a_n} \cdot \varphi_n(x)$$

In-class activity: Interpolation

Outline

Python, Numpy, and Matplotlib

Making Models with Polynomials

Making Models with Monte Carlo

Error, Accuracy and Convergence

Floating Point

Modeling the World with Arrays

The World in a Vector

What can Matrices Do?

Graphs

Sparsity

Norms and Errors

The 'Undo' Button for Linear Operations: LU

LU: Applications

Linear Algebra Applications

Interpolation

Repeating Linear Operations:

Eigenvalues and Steady States

Eigenvalues: Applications

Approximate Undo: SVD and

Least Squares

SVD: Applications

Solving Funny-Shaped Linear Systems

Data Fitting

Norms and Condition

Numbers

Low-Rank Approximation

Iteration and Convergence

Solving One Equation

Solving Many Equations

Finding the Best: Optimization in 1D

Optimization in n Dimensions

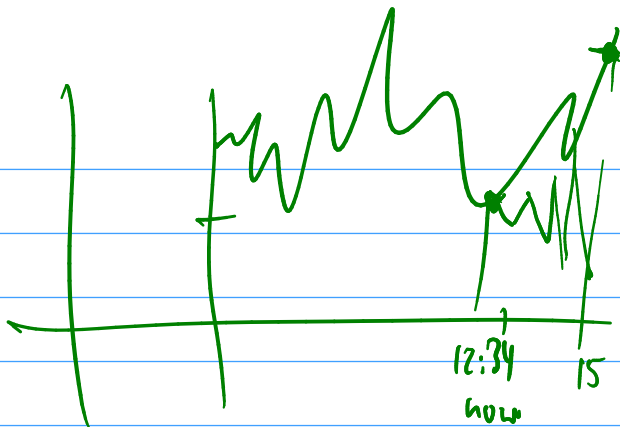
Randomness: Why?

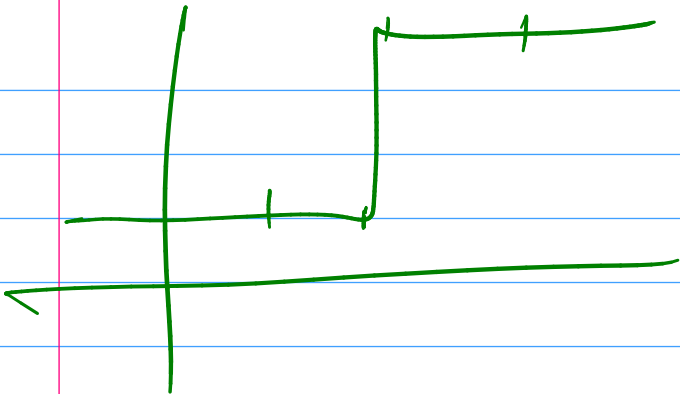
What types of problems can we solve with the help of random numbers?

We can compute (potentially) *complicated averages*.

- ▶ Where does 'the average' web surfer end up? (PageRank)
- ▶ How much is my stock portfolio/option going to be worth?
- ▶ How will my robot behave if there is measurement error?

4 2 3 1 6 1 2





Random Variables

What is a **random variable**?

A **random variable** X is a function that depends on 'the (random) state of the world'.

Example: X could be

- ▶ 'how much rain tomorrow?', or
- ▶ 'will my buttered bread land face-down?'

Idea: Since I don't know the entire state of the world (i.e. all the influencing factors), I can't know the value of X .

→ Next best thing: Say something about the *average* case.

To do that, I need to know how likely each individual value of X is. I need a **probability distribution**.

Probability Distributions

What kinds of probability distributions are there?

Demo: Plotting Distributions with Histograms