5. Taylor series and solving equations

Summary of last lecture

- Floating-point numbers:
  - `BigFloat`s
  - `Inf`, `NaN`
  - Sub-normal numbers

- We want to evaluate real functions
- Need to **approximate** using functions we **understand**
- Understand = can **compute**

- Polynomials are the only functions we understand

Outline for today

- Approximating functions: Taylor series
- Solving equations: Root finding
- Interactive plotting in Pluto

How can we *find* polynomial approximations?

- Given a function f on an interval $[a, b]$
- Given a tolerance ϵ
- Find a polynomial p that is close to f

How can we *find* polynomial approximations?

- Given a function f on an interval $[a, b]$
- Given a tolerance ϵ
- Find a polynomial p that is close to f

$$\|f - p\| < \epsilon$$

- We will see several methods during the course to do so

How can we *find* polynomial approximations?

- Given a function f on an interval $[a, b]$
- Given a tolerance ϵ
- Find a polynomial p that is close to f

$$\|f - p\| < \epsilon$$

- We will see several methods during the course to do so

- First technique: **Taylor series**
- Taylor series are key for many algorithms in the course

Refresh: Taylor series

- Suppose f is "nice" (smooth / analytic) near $a = 0$
- We can expand $f(x)$ as a **power series** in x:

$$f(x) = f_0 + f_1 x + f_2 x^2 + \cdots = \sum_{n=0}^{\infty} f_n x^n$$

Refresh: Taylor series

- Suppose f is "nice" (smooth / analytic) near $a = 0$
- We can expand $f(x)$ as a **power series** in x:

$$f(x) = f_0 + f_1 x + f_2 x^2 + \cdots = \sum_{n=0}^{\infty} f_n x^n$$

- By differentiating and evaluating at $x = 0$ we find

$$f_n = \frac{f^{(n)}(0)}{n!},$$

where $f^{(n)}(x)$ is the nth **derivative** of f

Example: exp

- We need to know how to calculate these derivatives
- For a general function f that can be complicated (see later)

Example: exp

- We need to know how to calculate these derivatives
- For a general function f that can be complicated (see later)

- For many functions we know the derivatives
- E.g. $\exp'(x) = \exp(x)$

Example: exp

- We need to know how to calculate these derivatives
- For a general function f that can be complicated (see later)

- For many functions we know the derivatives
- E.g. $\exp'(x) = \exp(x)$
- Alternatively: $f' = f$ gives **recurrence relations** for the coefficients f_n

Taylor series II

- We can expand around other points instead of 0
- Suppose we want to calculate $f(x)$ for x near a

Taylor series II

- We can expand around other points instead of 0
- Suppose we want to calculate $f(x)$ for x near a

- Set $h := x - a$ or $x = a + h$, with h **small** (near 0)
- Set $g(h) := f(a + h)$
- We know how to expand $g(h)$ in powers of h for h near 0

Taylor series III

$$f(x) = f(a+h) = g(h) = g_0 + g_1 h + g_2 h^2 + \cdots$$

Taylor series III

$$f(x) = f(a+h) = g(h) = g_0 + g_1 h + g_2 h^2 + \cdots$$

- We have $g_n = g^{(n)}(0) = f^{(n)}(a)$

Taylor series III

$$f(x) = f(a+h) = g(h) = g_0 + g_1 h + g_2 h^2 + \cdots$$

- We have $g_n = g^{(n)}(0) = f^{(n)}(a)$
- So, near a we have

$$f(a+h) = \sum_n \frac{1}{n!} f^{(n)}(a) \cdot h^n$$

Taylor series III

$$f(x) = f(a+h) = g(h) = g_0 + g_1 h + g_2 h^2 + \cdots$$

- We have $g_n = g^{(n)}(0) = f^{(n)}(a)$
- So, near a we have

$$f(a+h) = \sum_n \frac{1}{n!} f^{(n)}(a) \cdot h^n$$

- **Moral** $f(a+h)$ when h is *small* shouts "**Taylor**"

Implementing power series

- We cannot implement an infinite power series

Implementing power series

- We cannot implement an infinite power series
- Solution: **truncate** to finite number of terms
- We get a **polynomial**!

$$f_N(x) := \sum_{n=0}^{N} f_{[n]}(x-a)^n; \qquad f_{[n]} = f^{(n)}(a)/n!$$

Implementing power series

- We cannot implement an infinite power series
- Solution: **truncate** to finite number of terms
- We get a **polynomial**!

$$f_N(x) := \sum_{n=0}^{N} f_{[n]}(x-a)^n; \qquad f_{[n]} = f^{(n)}(a)/n!$$

- We have committed a **truncation error** by truncating

Implementing power series

- We cannot implement an infinite power series
- Solution: **truncate** to finite number of terms
- We get a **polynomial**!

$$f_N(x) := \sum_{n=0}^{N} f_{[n]}(x-a)^n; \qquad f_{[n]} = f^{(n)}(a)/n!$$

- We have committed a **truncation error** by truncating
- How can we estimate the size of the truncation error?

Estimating truncation error: Lagrange remainder

- Truncating to degree N leaves a **remainder** R_N:

$$f(x) = f_N(x) + R_N(f, x)$$

- With $R_N(f, x) := \sum_{N+1}^{\infty} f_{[n]}(x-a)^n$

Estimating truncation error: Lagrange remainder

- Truncating to degree N leaves a **remainder** R_N:

$$f(x) = f_N(x) + R_N(f, x)$$

- With $R_N(f, x) := \sum_{N+1}^{\infty} f_{[n]}(x-a)^n$

Estimating truncation error: Lagrange remainder

- Truncating to degree N leaves a **remainder** R_N:

$$f(x) = f_N(x) + R_N(f, x)$$

- With $R_N(f, x) := \sum_{N+1}^{\infty} f_{[n]}(x-a)^n$

- Lagrange gave a *finite* expression for the remainder:

$$R_N(f, x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-a)^{n+1}$$

where ξ is an *unknown* value in the interval $[a, x]$

Lagrange remainder II

- So if we can **bound** the $(n+1)$th derivative of f, we can estimate the size of the truncation error!

Lagrange remainder II

- So if we can **bound** the $(n+1)$th derivative of f, we can estimate the size of the truncation error!
- Note: There will also be **rounding error** since we use floating-point arithmetic

Lagrange remainder II

- So if we can **bound** the $(n+1)$th derivative of f, we can estimate the size of the truncation error!
- Note: There will also be **rounding error** since we use floating-point arithmetic
- Lagrange is a kind of generalized mean value theorem

Lagrange remainder II

- So if we can **bound** the $(n+1)$th derivative of f, we can estimate the size of the truncation error!
- Note: There will also be **rounding error** since we use floating-point arithmetic
- Lagrange is a kind of generalized mean value theorem
- Recall the **mean value theorem**:

If f is differentiable on $[a, b]$ then

$$\frac{f(b) - f(a)}{b - a} = f'(\xi)$$

- So $f(a+h) = f(a) + hf'(\xi)$

Collaborative exercise: Truncation error of exp(x)

Bounding the truncation error of $f = \exp$

1. Consider the function $f = \exp$ on the interval $[-1, 1]$ How can we bound the truncation error?
2. What about on the interval $[-b, b]$?
3. How could you ensure that the truncation error is below a given ϵ ?

Truncation error of exp(x) on $[-b, b]$

- Bounding each term, we find

$$|R_N(f, x)| \leq \frac{\mathsf{exp}(b)\, b^{n+1}}{(n+1)!}$$

Truncation error of exp(x) on $[-b, b]$

- Bounding each term, we find

$$|R_N(f, x)| \leq \frac{\exp(b)\, b^{n+1}}{(n+1)!}$$

- Using interval arithmetic it is possible to do this with *guaranteed bounds*
- The combination of a Taylor polynomial p and an interval I bounding $f - p$ is called a **Taylor model**
- We have an implemention in the `TaylorModels.jl` package

Solving equations in one variable

Solving equations in one variable: Root finding

- Suppose we have two functions f and g from $\mathbb{R} \to \mathbb{R}$
- We would like to **solve the equation**

$$f(x) = g(x)$$

Solving equations in one variable: Root finding

- Suppose we have two functions f and g from $\mathbb{R} \to \mathbb{R}$
- We would like to **solve the equation**

$$f(x) = g(x)$$

- I.e. find the special value(s) x^* such that $f(x^*) = g(x^*)$

Solving equations in one variable: Root finding

- Suppose we have two functions f and g from $\mathbb{R} \to \mathbb{R}$
- We would like to **solve the equation**

$$f(x) = g(x)$$

- I.e. find the special value(s) x^* such that $f(x^*) = g(x^*)$
- This occurs in many scientific and engineering problems

Collaborative exercise: Solving equations

Solving equations

1. Give examples from your discipline that require solving equations, with one or several variables
2. Restricting to one variable, how would you try to solve an equation $f(x) = g(x)$? (No fancy numerical methods, please; we'll come to those!)
3. Thinking back to calculating the square root $\sqrt{y}$ from problem set 1, which equation is being solved?
4. What property of the $\sqrt{\ }$ function allowed you to solve that?

Solving equations: Root finding

- It's convenient to modify the problem a bit

Solving equations: Root finding

- It's convenient to modify the problem a bit
- Move all terms to one side of the equation:

$$f(x) = g(x)$$

$$f(x) - g(x) = 0$$

Solving equations: Root finding

- It's convenient to modify the problem a bit
- Move all terms to one side of the equation:

$$f(x) = g(x)$$

$$f(x) - g(x) = 0$$

- So we have reduced the problem to solving

$$h(x) = 0$$

Solving equations: Root finding

- It's convenient to modify the problem a bit
- Move all terms to one side of the equation:

$$f(x) = g(x)$$

$$f(x) - g(x) = 0$$

- So we have reduced the problem to solving

$$h(x) = 0$$

- An x^* satisfying $h(x^*) = 0$ is called a **root** or **zero** of h

Collaborative exercise: Roots of polynomials

Roots (zeros) of polynomials

Suppose $p(x) = a_0 + a_1 x + \cdots + a_n x^n$ is a **polynomial** of degree n.

What do we know (from math) about the roots of p? Do they exist? How many are there?

1. What do we know if $n = 1$?
2. What do we know if $n = 2$?
3. What do we know if $n = 3$?
4. What do we know for general n?

Roots of polynomials

- The **Fundamental theorem of algebra** tells us that

  a degree-n polynomial has exactly n roots in $\mathbb{C}$

Roots of polynomials

- The **Fundamental theorem of algebra** tells us that

  a degree-n polynomial has exactly n roots in $\mathbb{C}$

- So we can factor p as

  $$p(x) = (x - x_1)^{m_1}(x - x_2)^{m_2} \cdots (x - x_k)^{m_k}$$

  where x_i are the roots and $\sum_{i=1}^{k} m_i = n$

- m_i is the **multiplicity** of root x_i

Roots of polynomials

- The **Fundamental theorem of algebra** tells us that

  a degree-n polynomial has exactly n roots in $\mathbb{C}$

- So we can factor p as

$$p(x) = (x - x_1)^{m_1}(x - x_2)^{m_2} \cdots (x - x_k)^{m_k}$$

  where x_i are the roots and $\sum_{i=1}^{k} m_i = n$

- m_i is the **multiplicity** of root x_i

- Visual proof!

Roots of polynomials II

- Some of the roots may be **multiple roots**
- E.g. $p = (x-1)^2(x^2-2)$ has roots 1, 1, $\sqrt{2}$ and $-\sqrt{2}$

Roots of polynomials II

- Some of the roots may be **multiple roots**
- E.g. $p = (x-1)^2(x^2-2)$ has roots 1, 1, $\sqrt{2}$ and $-\sqrt{2}$

- Roots may be complex even if all coefficients a_i are real
- E.g. x^2+1 has roots $x_{1,2} = \pm i$ and *no* real roots

Can we find roots of polynomial analytically?

- Suppose p is a degree-n polynomial with coefficients a_i
- $n = 1$ (linear): $ax + b = 0$; trivial solution
- $n = 2$: Quadratic formula for the roots in terms of the a_i
- $n = 3, 4$: complicated formulae exist
- Abel–Ruffini theorem:

*for $n \geq 5$, in general **no such formula exists**!*

Numerical methods for root finding

- We thus need *numerical* methods to find one or more roots
- Approximation algorithms for roots . . .

- They could be designed only for polynomials
- Or be more general
- We will look at some classical methods

Numerical methods for root finding

- We thus need *numerical* methods to find one or more roots
- Approximation algorithms for roots . . .

- They could be designed only for polynomials
- Or be more general
- We will look at some classical methods

- Note that interval arithmetic leads to methods with *guarantees*

Summary

- We can use Taylor series to approximate smooth functions in the vicinity of a point
- The Lagrange form of the remainder gives a computable bound

- Solving equations is the same as root finding
- We can solve polynomials exactly only up to degree 4
- So we need approximation algorithms for roots