# 3. Representing functions

# Last time

- Representing integers: Binary expansions
- Representing rationals: Defining a composite type
- Representing reals: Floating-point numbers

$$x = \pm 2^e(1+f); \qquad f = \sum_{n=1}^{d} 2^{-n} b_n$$

- Special rational numbers (denominator is power of 2)

# Goal for today

- How can we calculate elementary *functions*, e.g. `exp`
- Calculate $\Rightarrow$ **approximate**

# Floating-point arithmetic

- What happens if we multiply two floating-point numbers?
- Could do the exact computation with rationals
- But the lengths of the resulting numbers grows too fast
- Instead, we **round** the exact result to the *closest* floating-point number
- $+$, $-$, $*$, $/$ and $\sqrt{}$ have this **correct rounding** property
- NB: after several operations we may *lose accuracy*

# Elementary functions

- How approximate functions like $\exp(x)$ and $\sin(x)$?
- Want results that are as close as possible to true real result
- *Very hard* to guarantee correct rounding: **table-maker's dilemma**
- **Faithful rounding**: Return one of two neighbouring floating-point numbers

# Solving difficult problems

- $\exp(x)$ is difficult to work with
- **Idea**: Approximate a hard problem with a simpler problem
- Solve simpler problem
- Hope result gives approximate solution for hard problem
- How can we approximate $\exp(x)$ with a simpler problem?

# Polynomials

- Try to approximate exp$(x)$ with a **polynomial**
- Polynomials: Simplest functions to work with
- Polynomials: one foundations of numerical analysis
- $f(x) = a_0 + a_1\, x + \cdots + a_n x^n$ is a (univariate) **polynomial of degree** $n$
- *Finite* number of terms

# Polynomials II

- Can evaluate just using basic arithmetic
- (Almost) the *only* functions we can actually calculate!
- There is a more efficient evaluation method: Horner method (PS 1)
- Which other functions can we calculate?

# Can we approximate functions by polynomials?

- Weierstrass approximation theorem:
  *any* continuous *function on a finite interval can be approximated* ***uniformly*** *by a polynomial to within any given distance*

- "Uniformly" means that the maximum distance is bounded:

$$\max_{x \in [a,b]} \left| f(x) - p(x) \right| < \epsilon$$

- Or $\|f - p\|_\infty < \epsilon$
- This "uniform" condition fails for *infinite* intervals

## How can we find polynomial approximations?

- Will see several methods to *find* polynomials that approximate a function $f$
- Let's start with one that is motivated by looking at $y = \exp(x)$
- How is $\exp(x)$ defined?

## exp

- The function $\exp(x)$ is defined as a **power series**:

$$\exp(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

- Recall: the **factorial** $n! := 1 \times 2 \times \cdots \times n$
- Alternative definition: $\exp' = \exp$

  ($f' :=$ the derivative of $f$)

# Reminder: Taylor series

- Try to calculate $f(x)$ for $x$ near $0$
- Suppose $f$ is "nice" (smooth) so can expand as **power series**

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

- By differentiating and evaluating at $x = 0$, we see that

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

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

# Taylor series II

- We can expand around other points
- Suppose we want to calculate $f(x)$ for $x$ near $a$
- Set $h := x - a$ or $x = a + h$, so $h$ is **small**
- Set $g(h) := f(a + h)$.
- We know how to expand $g(h)$ in powers of $h$

# Taylor series III

- So $f(x) = f(a+h)$

$$= g(h) = g_0 + g_1 h + g_2 h^2 + \cdots \ldots$$

```
with $g_n = g^{(n)}(0) = f^{(n)}(a) =: f_n$
```

- So $f(x) = \sum_n \frac{1}{n!} f^{(n)}(a) \cdot (x-a)^n$

# Implementing power series

- How can we numerically implement a power series?
- As usual we *cannot*, since there are an infinite number of terms
- Solution: **truncate** to finite number of terms – get **polynomial**!

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

- Note that we have committed an error by truncating: **truncation error**

# Implementing $\exp(x)$

- Now have a polynomial, but don't just use previous method to evaluate – why?
- By taking more terms, expect result to approach true value
- How close and how fast?
- Do **numerical experiment**
- And try to estimate size of error term

# Estimating truncation error: Taylor theorem with Lagrange remainder

- Truncating to a polynomial of degree $n$ leaves a **remainder** $R_N$
- $f(x) = f_N(x) + R_N(f, x)$
- $R_N(f, x) := \sum_{N+1}^{\infty} f_n x^n$
- Fortunately there are finite expressions for this remainder
- Lagrange form:

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

- Here, $\xi$ is an *unknown* value in the interval $[a, x]$

# Lagrange remainder II

- **Bound** $(n+1)$ th derivative of $f$ to estimate size of truncation error
- NB: there may be **rounding error** too since calculate with floating-point arithmetic
- Lagrange result is kind of generalized mean value theorem
- **Mean value theorem**: If $f$ is differentiable on $[a, b]$ then

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

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

# Range reduction

- Where is the Taylor series of $f(x)$ valid?
- For some functions, e.g. exp$(x)$, it is valid **everywhere** in the complex plane!
- For e.g. log$(1 + x)$, finite **radius of convergence**
- To get fast convergence, want $x$ close to $0$ (why?),
- For larger values of $x$ we should relate $f(x)$ to $f(r)$ for some $r \in [-a, a]$

## Summary

- Key idea: **Approximate** functions using **polynomials**
- Can approximate as closely as desired (on finite interval) by taking higher degree
- Polynomials are the functions we best understand and can calculate
- One approximation method is via Taylor approximation