Overview

- Mathematical problem solutions
  - exact, symbolic, analytical
  - approximate, numerical
- Computer representations of $\mathbb{N}, \mathbb{Z}, \mathbb{R}, \mathbb{C}$
  - Unsigned integers: `UInt8`, `UInt16`, `UInt32`, `UInt64`
  - Signed integers: `Int8`, `Int16`, `Int32`, `Int64`
  - Floating point numbers: `Float32`, `Float64`, `Float128`
- Approximation sequences

- Analytic solutions: stated within the mathematical formalism
  - In $(\mathbb{R}, +, \cdot)$, $x^3 = 2$ has solution $x = 2^{1/3} = e^{(\ln 2)/3}$
  - In $(\mathbb{R}, +, \cdot)$, $y'(t) = e^t$ has solution $y(t) = e^t + C$
  - In $(\mathbb{R}, +, \cdot)$, $y'(t) = e^{-t^2}$ has solution $y(t) = \frac{\sqrt{\pi}}{2}\operatorname{erf}(t) + C = \int_0^t e^{-x^2}\mathrm{d}x + C$
- Though exact, in the sense of being stated within a given formalism, all the above require approximations to obtain quantitative answers

$$e^t = \frac{t^0}{0!} + \frac{t^1}{1!} + \frac{t^2}{2!} + \cdots + \frac{t^n}{n!} + \cdots$$

$$\operatorname{erf}(t) = \frac{2}{\sqrt{\pi}}\left(\frac{t}{1 \cdot 0!} - \frac{t^3}{3 \cdot 1!} + \frac{t^5}{5 \cdot 2!} - \cdots + (-1)^n \frac{t^{2n+1}}{(2n+1)\, n!} + \ldots\right)$$

- Approximations: successive terms of a sequence whose limit is the exact answer

$$x = 2^{1/3} = e^{(\ln 2)/3} = e^a, x_n = 1 + a + \frac{a^2}{2!} + \cdots + \frac{a^n}{n!}, \lim_{n\to\infty} x_n = 2^{1/3}$$

- Approximations need not be numerical, e.g., method of exhaustion.
- Multiple sequences can have the same limit

$$y_0 = 1, y_{n+1} = \frac{2}{3}\left(y_n + \frac{1}{y_n^2}\right), \lim_{n\to\infty} y_n = 2^{1/3}$$

- Numerical analysis
  - Devise approximation sequences for mathematical objects, $y'(t), \int y(t)\mathrm{d}t$
  - Determine the convergence behavior of the approximation sequences

- Mathematics defines $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$, e.g., $\mathbb{N}$ defined by counting by 1.
- Computers have finite memory, hence cannot exactly represent all numbers.
- $\mathbb{N}$ approximated by unsigned integers `UInt8`, `UInt16`, `UInt32`, `UInt64` using 8, 16, 32, 64 bits. `UInt8` can represent naturals from 0 to $2^8 - 1$.
- $\mathbb{Z}$ approximated by signed integers `Int8`, `Int16`, `Int32`, `Int64` using 8, 16, 32, 64 bits. `Int8` can represent integers from -2^7 to $2^7 - 1$.
- $\mathbb{Q}$ approximated (in some languages) by pair of signed integers
- $\mathbb{R}$ approximated by floating point numbers `Float32`, `Float64`, `Float128`

$$x = \pm 0.m_1 m_2 \ldots m_n \times 10^{b_1 b_2 \ldots b_p - 2^p}$$

  - sign bit: $\pm$
  - mantissa: $m_1 m_2 \ldots m_n$, $m_j \in \{0, 1\}$
  - biased exponent: $b_1 b_2 \ldots b_p$, $b_j \in \{0, 1\}$

- $\mathbb{C}$ approximated by pair of floating point numbers

- Number approximations do not necessarily satisfy properties of $\mathbb{N}, \mathbb{Z}, \mathbb{R}$
- Examples:
  - In UInt8: 150+200 cannot be represented, *overflow*
  - In Float32, the series

$$S_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}, T_n = \frac{1}{n} + \frac{1}{n-1} + \cdots + 1$$

    with displayed ordering of terms have different values for large n

- Quantify precision of floating point system by *machine epsilon* ϵ, smallest number of form $\epsilon = 2^k \in \mathbb{F}$ that satisfies $1 + \epsilon \neq 1$.

```
∴ eps=1.0;
```

```
∴ while (1.0+0.5*eps != 1.0)
    global eps;
    eps=0.5*eps;
  end
```

- As mentioned, different sequences can have the same limit

$$x_1 = 1.5, x_{n+1} = \frac{1}{2}\left(x_n + \frac{2}{x_n}\right), \lim_{n\to\infty} x_n = \sqrt{2}$$

$$\sqrt{2} = 1 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \ddots}}}$$

- The continued fraction can be evaluated through

$$z_1 = 2, z_{n+1} = 2 + \frac{1}{z_n}, y_n = 1 + \frac{1}{z_n}, \lim_{n\to\infty} y_n = \sqrt{2}$$

- A natural question is: which sequence to choose? Considerations involve accuracy attained for given n, computational effort for given n.