MA579H Scientific Computing

Lectures 2 & 3: Errors and Floating-point Arithmetic

Rafikul Alam
Department of Mathematics
IIT Guwahati

Lecture outline

- Sources of errors
- Floating-point arithmetic
- Rounding error

Sources of errors

Example: Suppose we wish to compute surface area of Earth using formula $S = 4\pi r^2$. This involves several approximations:

Sources of errors

Example: Suppose we wish to compute surface area of Earth using formula $S = 4\pi r^2$. This involves several approximations:

- Modeling error: Earth is modeled as a sphere, idealizing its true shape.
- Measurement error: Computation of the radius r is based on empirical measurements.

Sources of errors

Example: Suppose we wish to compute surface area of Earth using formula $S = 4\pi r^2$. This involves several approximations:

- Modeling error: Earth is modeled as a sphere, idealizing its true shape.
- Measurement error: Computation of the radius r is based on empirical measurements.
- Truncation error: Computation of π requires truncation of infinite process.
- Rounding error: Input data and results of arithmetic operations are rounded in computer.

Sources of errors

Before computation:

- Modeling error.
- Empirical measurement errors.

Sources of errors

Before computation:

- Modeling error.
- Empirical measurement errors.

During computation:

- Truncation or discretization error
  - infinite $\longrightarrow$ finite
  - differential or integral $\longrightarrow$ algebraic
  - nonlinear $\longrightarrow$ linear
  - complicated $\longrightarrow$ simple
- Rounding error.

Sources of errors

Before computation:

- Modeling error.
- Empirical measurement errors.

During computation:

- Truncation or discretization error
  - infinite $\longrightarrow$ finite
  - differential or integral $\longrightarrow$ algebraic
  - nonlinear $\longrightarrow$ linear
  - complicated $\longrightarrow$ simple
- Rounding error.

Scientific computing analyzes errors that arise during computation. A small error in the input may be amplified

- by a problem (bad problem)
- by an algorithm (bad algorithm) during computation.

Errors

Two common measures of error are

$$\text{Absolute Error} := \|\text{True Value} - \text{Approx. Value}\|$$

$$\text{Relative Error} := \frac{\text{Absolute Error}}{\|\text{True Value}\|}$$

Errors

Two common measures of error are

$$\text{Absolute Error} := \|\text{True Value} - \text{Approx. Value}\|$$

$$\text{Relative Error} := \frac{\text{Absolute Error}}{\|\text{True Value}\|}$$

But

- true value is usually unknown,
- so we estimate or bound errors rather than compute them exactly.

Errors

Two common measures of error are

$$\text{Absolute Error} := \|\text{True Value} - \text{Approx. Value}\|$$

$$\text{Relative Error} := \frac{\text{Absolute Error}}{\|\text{True Value}\|}$$

But

- true value is usually unknown,
- so we estimate or bound errors rather than compute them exactly.

Which measure of error we consider depends on the purpose and the sensitivity of the process under consideration.

Errors

Two common measures of error are

$$\text{Absolute Error} := \|\text{True Value} - \text{Approx. Value}\|$$

$$\text{Relative Error} := \frac{\text{Absolute Error}}{\|\text{True Value}\|}$$

But

- true value is usually unknown,
- so we estimate or bound errors rather than compute them exactly.

Which measure of error we consider depends on the purpose and the sensitivity of the process under consideration.

- Astronomers may be interested in relative errors.
- Missile technologists may be interested in absolute errors.

Errors

Two common measures of error are

$$\text{Absolute Error} := \|\text{True Value} - \text{Approx. Value}\|$$

$$\text{Relative Error} := \frac{\text{Absolute Error}}{\|\text{True Value}\|}$$

But

- true value is usually unknown,
- so we estimate or bound errors rather than compute them exactly.

Which measure of error we consider depends on the purpose and the sensitivity of the process under consideration.

- Astronomers may be interested in relative errors.
- Missile technologists may be interested in absolute errors.

Relative error is considered for analysis of rounding errors.

Anomalous arithmetic

Arithmetic operations on computers are inexact and can produce surprising results. For example, MATLAB produces

$$
\begin{aligned}
(\frac{4}{3} - 1) * 3 - 1 &= -2.2204 \times 10^{-16} \\
5 \times \frac{(1 + \exp(-50)) - 1}{(1 + \exp(-50)) - 1} &= \textbf{NaN} \\
\frac{\log(\exp(750))}{100} &= \textbf{Inf}
\end{aligned}
$$

Normalized floating-point representation

Let $x \in \mathbb{R}$ be nonzero and $\beta > 1$ be an integer. Then we have the normalized floating-point representation

$$x = \pm(\cdot d_1 d_2 \cdots d_t \cdots)_\beta \times \beta^e, \; 0 \leq d_i < \beta, \;\; d_1 \neq 0, \;\; e \in \mathbb{Z}.$$

Here β is called the base, e is called the exponent and $f := (\cdot d_1 d_2 \cdots d_t \cdots)_\beta$ is called mantissa or fraction and given by

Normalized floating-point representation

Let $x \in \mathbb{R}$ be nonzero and $\beta > 1$ be an integer. Then we have the normalized floating-point representation

$$x = \pm(\cdot d_1 d_2 \cdots d_t \cdots)_\beta \times \beta^e, \ 0 \leq d_i < \beta, \ \ d_1 \neq 0, \ \ e \in \mathbb{Z}.$$

Here β is called the base, e is called the exponent and $f := (\cdot d_1 d_2 \cdots d_t \cdots)_\beta$ is called mantissa or fraction and given by

$$f = (\cdot d_1 d_2 \cdots d_t \cdots)_\beta = \sum_{j=1}^{\infty} \frac{d_j}{\beta^j}.$$

Note that $0 < f \leq 1$ and $x = \pm f \times \beta^e$ is a unique representation. The number of digits allowed in f is called precision of the floating-point representation, which in the present case is infinite.

Floating-Point System

Computers use a floating-point representation with base $\beta = 2$. The set of numbers represented on a computer is called a floating-point system.

Floating-Point System

Computers use a floating-point representation with base $\beta = 2$. The set of numbers represented on a computer is called a floating-point system.

A floating-point system $F(\beta, t, e_{\min}, e_{\max})$ is a finite set given by

$$\{\pm(\cdot d_1 d_2 \cdots d_t)_\beta \times \beta^e,\ 0 \leq d_i < \beta,\ \ d_1 \neq 0,\ e_{\min} \leq e \leq e_{\max}\} \cup \{0\}.$$

Floating-Point System

Computers use a floating-point representation with base $\beta = 2$. The set of numbers represented on a computer is called a floating-point system.

A floating-point system $F(\beta, t, e_{\min}, e_{\max})$ is a finite set given by

$$\{\pm(\cdot d_1 d_2 \cdots d_t)_\beta \times \beta^e,\ 0 \leq d_i < \beta,\ \ d_1 \neq 0,\ e_{\min} \leq e \leq e_{\max}\} \cup \{0\}.$$

β : base of the floating-point system

t : precision (number of digits in the mantissa) of the floating-point system

Floating-Point System

Computers use a floating-point representation with base $\beta = 2$. The set of numbers represented on a computer is called a floating-point system.

A floating-point system $F(\beta, t, e_{\min}, e_{\max})$ is a finite set given by

$$\{\pm(\cdot d_1 d_2 \cdots d_t)_\beta \times \beta^e,\ 0 \leq d_i < \beta,\ \ d_1 \neq 0,\ e_{\min} \leq e \leq e_{\max}\} \cup \{0\}.$$

β : base of the floating-point system

t : precision (number of digits in the mantissa) of the floating-point system

$e_{\min}$: lowest exponent of the floating-point system

$e_{\max}$: highest exponent of the floating-point system

Floating-Point System

Computers use a floating-point representation with base $\beta = 2$. The set of numbers represented on a computer is called a floating-point system.

A floating-point system $F(\beta, t, e_{\min}, e_{\max})$ is a finite set given by

$$\{\pm(\cdot d_1 d_2 \cdots d_t)_\beta \times \beta^e,\ 0 \leq d_i < \beta,\ \ d_1 \neq 0,\ e_{\min} \leq e \leq e_{\max}\} \cup \{0\}.$$

β : base of the floating-point system

t : precision (number of digits in the mantissa) of the floating-point system

$e_{\min}$: lowest exponent of the floating-point system

$e_{\max}$: highest exponent of the floating-point system

The fraction $f := (\cdot d_1 d_2 \cdots d_t)_\beta = \sum_{j=1}^{t} \frac{d_j}{\beta^j} \leq 1 - \beta^{-t} < 1.$

Example: $(\cdot 101)_2 \times 2^2 = (1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}) \times 2^2 = 5/2.$

Floating-Point System

Computers use a floating-point representation with base $\beta = 2$. The set of numbers represented on a computer is called a floating-point system.

A floating-point system $F(\beta, t, e_{\min}, e_{\max})$ is a finite set given by

$$\{\pm(\cdot d_1 d_2 \cdots d_t)_\beta \times \beta^e,\ 0 \le d_i < \beta,\ \ d_1 \neq 0,\ e_{\min} \le e \le e_{\max}\} \cup \{0\}.$$

β : base of the floating-point system

t : precision (number of digits in the mantissa) of the floating-point system

$e_{\min}$: lowest exponent of the floating-point system

$e_{\max}$: highest exponent of the floating-point system

The fraction $f := (\cdot d_1 d_2 \cdots d_t)_\beta = \sum_{j=1}^{t} \frac{d_j}{\beta^j} \le 1 - \beta^{-t} < 1.$

Example: $(\cdot 101)_2 \times 2^2 = (1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}) \times 2^2 = 5/2.$

Note that $\#F(\beta, t, e_{\min}, e_{\max}) = 2(\beta - 1)\beta^{t-1} \times (e_{\max} - e_{\min} + 1) + 1.$

Floating-Point System

Underflow threshold (smallest positive floating point number):

- $\texttt{realmin} := (\cdot 10\cdots 0)_\beta \times \beta^{e_{\min}} = \beta^{e_{\min}-1}$

Overflow threshold (largest positive floating point number):

- $\texttt{realmax} := (\cdot(\beta-1)\cdots(\beta-1))_\beta \times \beta^{e_{\max}} = (1-\beta^{-t})\beta^{e_{\max}}$

Floating-Point System

Underflow threshold (smallest positive floating point number):

- $\texttt{realmin} := (\cdot 10\cdots 0)_\beta \times \beta^{e_{\min}} = \beta^{e_{\min}-1}$

Overflow threshold (largest positive floating point number):

- $\texttt{realmax} := (\cdot(\beta-1)\cdots(\beta-1))_\beta \times \beta^{e_{\max}} = (1-\beta^{-t})\beta^{e_{\max}}$

Machine epsilon/precision: **eps** $:= \beta^{1-t}$.

- **eps** $= (\cdot 10\cdots 01)_\beta \times \beta^1 - (\cdot 10\cdots 0)_\beta \times \beta^1 = \beta^{1-t}$

Floating-Point System

Underflow threshold (smallest positive floating point number):

- $\texttt{realmin} := (\cdot 10\cdots 0)_\beta \times \beta^{e_{\min}} = \beta^{e_{\min}-1}$

Overflow threshold (largest positive floating point number):

- $\texttt{realmax} := (\cdot(\beta-1)\cdots(\beta-1))_\beta \times \beta^{e_{\max}} = (1-\beta^{-t})\beta^{e_{\max}}$

Machine epsilon/precision: **eps** $:= \beta^{1-t}$.

- **eps** $= (\cdot 10\cdots 01)_\beta \times \beta^1 - (\cdot 10\cdots 0)_\beta \times \beta^1 = \beta^{1-t}$

For MATLAB we have the following

	Binary	Decimal
eps	2^{-52}	2.2204×10^{-16}
realmin	2^{-1022}	2.2251×10^{-308}
realmax	$(2-\texttt{eps}) \times 2^{1023}$	1.7977×10^{308}

IEEE single precision floating-point representation

IEEE specifies $F(2, t, e_{\min}, e_{\max})$ as follows.

sign (1 bit)	exponent (8 bits)	mantissa (23 bits)

Table : Single precision 32 bits

IEEE single precision floating-point representation

IEEE specifies $F(2, t, e_{\min}, e_{\max})$ as follows.

sign (1 bit)	exponent (8 bits)	mantissa (23 bits)

Table : Single precision 32 bits

$\#$ Exponents: $2^8 = 256$ Largest exponent: $2^8 - 1 = 255$
Exponent range: $0 \leq b \leq 255$ and $-127 \leq e \leq 128$

$$x = (-1)^s \times (1 \cdot d_1 \cdots d_{23})_2 \times 2^{b-127} \text{ where } e = b - 127.$$

Effective exponent range: $-126 \leq e \leq 127$ as $e = -127$ is reserved for ± 0 and $e = 128$ for $\pm\texttt{inf}$ and $\texttt{NaN}$

IEEE single precision floating-point representation

IEEE specifies $F(2, t, e_{\min}, e_{\max})$ as follows.

sign (1 bit)	exponent (8 bits)	mantissa (23 bits)

Table : Single precision 32 bits

\# Exponents: $2^8 = 256$ Largest exponent: $2^8 - 1 = 255$
Exponent range: $0 \leq b \leq 255$ and $-127 \leq e \leq 128$

$$x = (-1)^s \times (1 \cdot d_1 \cdots d_{23})_2 \times 2^{b-127} \text{ where } e = b - 127.$$

Effective exponent range: $-126 \leq e \leq 127$ as $e = -127$ is reserved for ± 0 and $e = 128$ for $\pm\texttt{inf}$ and $\texttt{NaN}$

$\texttt{eps} = 2^{-23} \approx 1.192 \times 10^{-7}$, $\texttt{realmin} = 2^{-126} \approx 1.175 \times 10^{-38}$, $\texttt{realmax} = (2 - \texttt{eps}) \times 2^{127} \approx 3.403 \times 10^{38}$

IEEE double precision floating-point representation

IEEE specifies $F(2, t, e_{\min}, e_{\max})$ as follows.

sign (1 bit)	exponent (11 bits)	mantissa (52 bits)

Table : Double precision 64 bits

IEEE double precision floating-point representation

IEEE specifies $F(2, t, e_{\min}, e_{\max})$ as follows.

sign (1 bit)	exponent (11 bits)	mantissa (52 bits)

Table : Double precision 64 bits

\# Exponents: $2^{11} = 2048$ Largest exponent: $2^{11} - 1 = 2047$
Exponent range: $0 \leq b \leq 2047$ and $-1023 \leq e \leq 1024$

$$x = (-1)^s \times (1 \cdot d_1 \cdots d_{52})_2 \times 2^{b-1023} \text{ where } e = b - 1023.$$

Effective exponent range: $-1022 \leq e \leq 1023$ as $e = -1023$ is reserved for ± 0 and $e = 1024$ for $\pm\texttt{inf}$ and $\texttt{NaN}$

IEEE double precision floating-point representation

IEEE specifies $F(2, t, e_{\min}, e_{\max})$ as follows.

sign (1 bit)	exponent (11 bits)	mantissa (52 bits)

Table : Double precision 64 bits

\# Exponents: $2^{11} = 2048$ Largest exponent: $2^{11} - 1 = 2047$
Exponent range: $0 \leq b \leq 2047$ and $-1023 \leq e \leq 1024$

$$x = (-1)^s \times (1 \cdot d_1 \cdots d_{52})_2 \times 2^{b-1023} \text{ where } e = b - 1023.$$

Effective exponent range: $-1022 \leq e \leq 1023$ as $e = -1023$ is reserved for ± 0 and $e = 1024$ for $\pm$`inf` and `NaN`

$\texttt{eps} = 2^{-52} \approx 2.2204 \times 10^{-16}$, $\texttt{realmin} = 2^{-1022} \approx 2.2251 \times 10^{-308}$, $\texttt{realmax} = (2 - \texttt{eps}) \times 2^{1023} \approx 1.7977 \times 10^{308}$

Gap between floating-point numbers

Let $x \in F(\beta, t, e_{\min}, e_{\max})$ be given by $x = (\cdot d_1 d_2 \cdots d_t)_\beta \times \beta^e$.

Define $\mathrm{ulp}(x) := \beta^{e-t}$. Then $\mathrm{next}(x) := x + \mathrm{ulp}(x)$ is the next floating point number larger than x, that is, $\mathrm{next}(x)$ is the smallest floating-point number larger than x. Hence x and $\mathrm{next}(x)$ are consecutive floating-point numbers.

Gap between floating-point numbers

Let $x \in F(\beta, t, e_{\min}, e_{\max})$ be given by $x = (\cdot d_1 d_2 \cdots d_t)_\beta \times \beta^e$.

Define $\mathrm{ulp}(x) := \beta^{e-t}$. Then $\mathrm{next}(x) := x + \mathrm{ulp}(x)$ is the next floating point number larger than x, that is, $\mathrm{next}(x)$ is the smallest floating-point number larger than x. Hence x and $\mathrm{next}(x)$ are consecutive floating-point numbers.

This shows that $\mathrm{next}(x) - x = \mathrm{ulp}(x) = \beta^{e-t}$. So, if x is a large number then e is large which implies large gap between x and $\mathrm{next}(x)$. Hence the gap between consecutive floating-point number is nonuniform.

Gap between floating-point numbers

Let $x \in F(\beta, t, e_{\min}, e_{\max})$ be given by $x = (\cdot d_1 d_2 \cdots d_t)_\beta \times \beta^e$.

Define $\mathrm{ulp}(x) := \beta^{e-t}$. Then $\mathrm{next}(x) := x + \mathrm{ulp}(x)$ is the next floating point number larger than x, that is, $\mathrm{next}(x)$ is the smallest floating-point number larger than x. Hence x and $\mathrm{next}(x)$ are consecutive floating-point numbers.

This shows that $\mathrm{next}(x) - x = \mathrm{ulp}(x) = \beta^{e-t}$. So, if x is a large number then e is large which implies large gap between x and $\mathrm{next}(x)$. Hence the gap between consecutive floating-point number is nonuniform.

Example: Consider $x \in F(10, 4, -30, 30)$ given by $x = 10^{-9} = .1000 \times 10^{-8}$. Then $\mathrm{ulp}(x) = 10^{-8-4} = 10^{-12}$ and $\mathrm{next}(x) - x = 10^{-12}$.

Gap between floating-point numbers

Let $x \in F(\beta, t, e_{\min}, e_{\max})$ be given by $x = (\cdot d_1 d_2 \cdots d_t)_\beta \times \beta^e$.

Define $\mathrm{ulp}(x) := \beta^{e-t}$. Then $\mathrm{next}(x) := x + \mathrm{ulp}(x)$ is the next floating point number larger than x, that is, $\mathrm{next}(x)$ is the smallest floating-point number larger than x. Hence x and $\mathrm{next}(x)$ are consecutive floating-point numbers.

This shows that $\mathrm{next}(x) - x = \mathrm{ulp}(x) = \beta^{e-t}$. So, if x is a large number then e is large which implies large gap between x and $\mathrm{next}(x)$. Hence the gap between consecutive floating-point number is nonuniform.

Example: Consider $x \in F(10, 4, -30, 30)$ given by $x = 10^{-9} = .1000 \times 10^{-8}$. Then $\mathrm{ulp}(x) = 10^{-8-4} = 10^{-12}$ and $\mathrm{next}(x) - x = 10^{-12}$.

Next, consider $y = 10^{15} = .1000 \times 10^{16}$. Then $\mathrm{ulp}(y) = 10^{16-4} = 10^{12}$ and $\mathrm{next}(y) - y = 10^{12}$.

Relative gap between floating-point numbers

Let $x \in F(\beta, t, e_{\min}, e_{\max})$ be given by $x = (\cdot d_1 d_2 \cdots d_t)_\beta \times \beta^e$. Then $\mathrm{next}(x) = x + \mathrm{ulp}(x) = x + \beta^{e-t}$ implies that the relative gap

$$\frac{\mathrm{next}(x) - x}{x} = \frac{\beta^{e-t}}{(.d_1 \cdots d_t)_\beta \times \beta^e} \leq \beta^{1-t} = \mathrm{eps}.$$

Hence the relative gap between two consecutive floating-point numbers is almost uniform and the relative gap is at most eps.

Relative gap between floating-point numbers

Let $x \in F(\beta, t, e_{\min}, e_{\max})$ be given by $x = (\cdot d_1 d_2 \cdots d_t)_\beta \times \beta^e$. Then $\mathrm{next}(x) = x + \mathrm{ulp}(x) = x + \beta^{e-t}$ implies that the relative gap

$$\frac{\mathrm{next}(x) - x}{x} = \frac{\beta^{e-t}}{(.d_1 \cdots d_t)_\beta \times \beta^e} \leq \beta^{1-t} = \mathrm{eps}.$$

Hence the relative gap between two consecutive floating-point numbers is almost uniform and the relative gap is at most eps.

Example: Consider $x \in F(10, 4, -30, 30)$ given by $x = 10^{-9} = .1000 \times 10^{-8}$. Then $\mathrm{ulp}(x) = 10^{-8-4} = 10^{-12}$ and $\mathrm{next}(x) - x = 10^{-12}$.

Relative gap between floating-point numbers

Let $x \in F(\beta, t, e_{\min}, e_{\max})$ be given by $x = (\cdot d_1 d_2 \cdots d_t)_\beta \times \beta^e$. Then $\mathrm{next}(x) = x + \mathrm{ulp}(x) = x + \beta^{e-t}$ implies that the relative gap

$$\frac{\mathrm{next}(x) - x}{x} = \frac{\beta^{e-t}}{(.d_1 \cdots d_t)_\beta \times \beta^e} \leq \beta^{1-t} = \mathrm{eps}.$$

Hence the relative gap between two consecutive floating-point numbers is almost uniform and the relative gap is at most eps.

Example: Consider $x \in F(10, 4, -30, 30)$ given by $x = 10^{-9} = .1000 \times 10^{-8}$. Then $\mathrm{ulp}(x) = 10^{-8-4} = 10^{-12}$ and $\mathrm{next}(x) - x = 10^{-12}$. Note, however, that $\dfrac{\mathrm{next}(x) - x}{x} = \dfrac{10^{-12}}{10^{-9}} = 10^{-3} = \mathrm{eps}.$

Relative gap between floating-point numbers

Let $x \in F(\beta, t, e_{\min}, e_{\max})$ be given by $x = (\cdot d_1 d_2 \cdots d_t)_\beta \times \beta^e$. Then $\mathrm{next}(x) = x + \mathrm{ulp}(x) = x + \beta^{e-t}$ implies that the relative gap

$$\frac{\mathrm{next}(x) - x}{x} = \frac{\beta^{e-t}}{(.d_1 \cdots d_t)_\beta \times \beta^e} \leq \beta^{1-t} = \mathrm{eps}.$$

Hence the relative gap between two consecutive floating-point numbers is almost uniform and the relative gap is at most eps.

Example: Consider $x \in F(10, 4, -30, 30)$ given by $x = 10^{-9} = .1000 \times 10^{-8}$. Then $\mathrm{ulp}(x) = 10^{-8-4} = 10^{-12}$ and $\mathrm{next}(x) - x = 10^{-12}$. Note, however, that $\dfrac{\mathrm{next}(x) - x}{x} = \dfrac{10^{-12}}{10^{-9}} = 10^{-3} = \mathrm{eps}.$

Next, consider $y = 10^{15} = .1000 \times 10^{16}$. Then $\mathrm{ulp}(y) = 10^{16-4} = 10^{12}$ and

$\mathrm{next}(y) - y = 10^{12}$.

Relative gap between floating-point numbers

Let $x \in F(\beta, t, e_{\min}, e_{\max})$ be given by $x = (\cdot d_1 d_2 \cdots d_t)_\beta \times \beta^e$. Then $\mathrm{next}(x) = x + \mathrm{ulp}(x) = x + \beta^{e-t}$ implies that the relative gap

$$\frac{\mathrm{next}(x) - x}{x} = \frac{\beta^{e-t}}{(.d_1 \cdots d_t)_\beta \times \beta^e} \leq \beta^{1-t} = \mathrm{eps}.$$

Hence the relative gap between two consecutive floating-point numbers is almost uniform and the relative gap is at most eps.

Example: Consider $x \in F(10, 4, -30, 30)$ given by $x = 10^{-9} = .1000 \times 10^{-8}$. Then $\mathrm{ulp}(x) = 10^{-8-4} = 10^{-12}$ and $\mathrm{next}(x) - x = 10^{-12}$. Note, however, that $\dfrac{\mathrm{next}(x) - x}{x} = \dfrac{10^{-12}}{10^{-9}} = 10^{-3} = \mathrm{eps}.$

Next, consider $y = 10^{15} = .1000 \times 10^{16}$. Then $\mathrm{ulp}(y) = 10^{16-4} = 10^{12}$ and $\mathrm{next}(y) - y = 10^{12}$.However, we have $\dfrac{\mathrm{next}(y) - y}{y} = \dfrac{10^{12}}{10^{15}} = 10^{-3} = \mathrm{eps}.$ ■

Rounding

A number $x \in \mathbb{R}$ can be represented by a number in $F(\beta, t, e_{\min}, e_{\max})$ provided $\texttt{realmin} \leq |x| \leq \texttt{realmax}$.

Rounding

A number $x \in \mathbb{R}$ can be represented by a number in $F(\beta, t, e_{\min}, e_{\max})$ provided $\texttt{realmin} \leq |x| \leq \texttt{realmax}$.

Let $\mathrm{fl}(x)$ denote the floating-point representation of x, that is,

$$\mathrm{fl} : \mathbb{R} \to F(\beta, t, L, U), \; x \longmapsto \mathrm{fl}(x).$$

How to define $\mathrm{fl}(x)$?

Rounding

A number $x \in \mathbb{R}$ can be represented by a number in $F(\beta, t, e_{\min}, e_{\max})$ provided $\texttt{realmin} \leq |x| \leq \texttt{realmax}$.

Let $\mathrm{fl}(x)$ denote the floating-point representation of x, that is,

$$\mathrm{fl} : \mathbb{R} \to F(\beta, t, L, U), \ x \longmapsto \mathrm{fl}(x).$$

How to define $\mathrm{fl}(x)$?

Let $x = (\cdot d_1 d_2 \cdots d_t \cdots)_\beta \times \beta^e$. Set $x_L := (\cdot d_1 d_2 \cdots d_t)_\beta \times \beta^e$ and $x_R := x_L + (\cdot 0 \cdots 01)_\beta \times \beta^e = x_L + \beta^{e-t}$. Then x_R is next floating-point number larger than x_L and $x_L \leq x \leq x_R$.

Rounding

A number $x \in \mathbb{R}$ can be represented by a number in $F(\beta, t, e_{\min}, e_{\max})$ provided $\texttt{realmin} \le |x| \le \texttt{realmax}$.

Let $\mathrm{fl}(x)$ denote the floating-point representation of x, that is,

$$\mathrm{fl} : \mathbb{R} \to F(\beta, t, L, U), \; x \longmapsto \mathrm{fl}(x).$$

How to define $\mathrm{fl}(x)$?

Let $x = (\cdot d_1 d_2 \cdots d_t \cdots)_\beta \times \beta^e$. Set $x_L := (\cdot d_1 d_2 \cdots d_t)_\beta \times \beta^e$ and $x_R := x_L + (\cdot 0 \cdots 01)_\beta \times \beta^e = x_L + \beta^{e-t}$. Then x_R is next floating-point number larger than x_L and $x_L \le x \le x_R$.

$$\text{Define } \mathrm{fl}(x) := \begin{cases} x_L, & \text{round down} \\ x_R, & \text{round up} \\ x_L \text{ or } x_R \text{ whichever is closer to } x, & \text{round to nearest} \end{cases}$$

Rounding error

Unit roundoff: $$\mathbf{u} := \begin{cases} \beta^{1-t} & \text{round down,} \\ \frac{1}{2}\beta^{1-t}, & \text{round to nearest} \end{cases}$$

Rounding error

Unit roundoff: $\mathbf{u} := \begin{cases} \beta^{1-t} & \text{round down,} \\ \frac{1}{2}\beta^{1-t}, & \text{round to nearest} \end{cases}$

Theorem: Let $\mathrm{fl} : \mathbb{R} \to F(\beta, t, L, U),\; x \longmapsto \mathrm{fl}(x)$. Then

$$\mathrm{fl}(x) = x(1+\delta) \text{ with } |\delta| \leq \mathbf{u}.$$

Rounding error

Unit roundoff: $\mathbf{u} := \begin{cases} \beta^{1-t} & \text{round down,} \\ \frac{1}{2}\beta^{1-t}, & \text{round to nearest} \end{cases}$

Theorem: Let $\mathrm{fl} : \mathbb{R} \to F(\beta, t, L, U),\ x \longmapsto \mathrm{fl}(x)$. Then

$$\mathrm{fl}(x) = x(1+\delta) \text{ with } |\delta| \leq \mathbf{u}.$$

Proof: Let $x = (\cdot d_1 d_2 \cdots d_t \cdots)_\beta \times \beta^e$. Set $\delta := \frac{\mathrm{fl}(x)-x}{x}$. Then $\mathrm{fl}(x) = x(1+\delta)$. Note that $|\mathrm{fl}(x) - x| \leq \beta^{e-t}$ for chopping, and $|\mathrm{fl}(x) - x| \leq \beta^{e-t}/2$ for round to nearest. Also $x \geq \beta^{e-1}$.

Rounding error

Unit roundoff: $\mathbf{u} := \begin{cases} \beta^{1-t} & \text{round down,} \\ \frac{1}{2}\beta^{1-t}, & \text{round to nearest} \end{cases}$

Theorem: Let $\mathrm{fl} : \mathbb{R} \to F(\beta, t, L, U),\ x \longmapsto \mathrm{fl}(x)$. Then

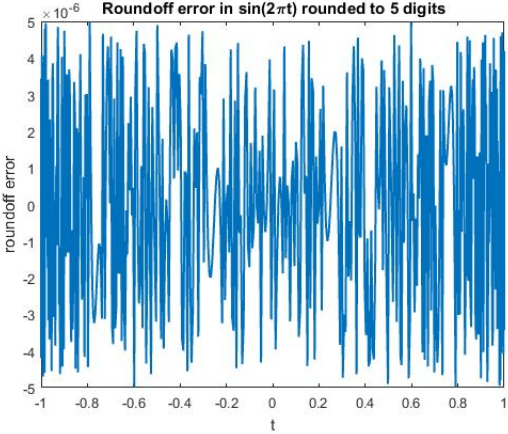
$$\mathrm{fl}(x) = x(1+\delta) \text{ with } |\delta| \leq \mathbf{u}.$$

Proof: Let $x = (\cdot d_1 d_2 \cdots d_t \cdots)_\beta \times \beta^e$. Set $\delta := \frac{\mathrm{fl}(x)-x}{x}$. Then $\mathrm{fl}(x) = x(1+\delta)$. Note that $|\mathrm{fl}(x) - x| \leq \beta^{e-t}$ for chopping, and $|\mathrm{fl}(x) - x| \leq \beta^{e-t}/2$ for round to nearest. Also $x \geq \beta^{e-1}$.
Hence

$$|\delta| \leq \frac{|\mathrm{fl}(x) - x|}{|x|} \leq \begin{cases} \beta^{1-t} & \text{round down,} \\ \frac{1}{2}\beta^{1-t}, & \text{round to nearest.} \end{cases}$$

Rounding error

Let $f(t) := \sin(2\pi t)$ for $t \in [0, 1]$. Consider $F(10, 5, -10, 10)$. Then `t = 0:.001:1; tt = sin(2*pi*t); rt = round(tt, 5);` `roundoff = tt-rt; plot(tt, roundoff, 'LineWidth',1.5)` produces the following



Floating-point arithmetic

Arithmetic model: If $x, y \in F(\beta, t, e_{\min}, e_{\max})$ and $\oplus \in \{+, -, \times, /\}$ and $\mathbf{u}$ is the unit roundoff then

$$\mathrm{fl}(x \oplus y) = (x \oplus y)(1 + \delta), \text{ where } |\delta| \le \mathbf{u}.$$

Floating-point arithmetic

Arithmetic model: If $x, y \in F(\beta, t, e_{\min}, e_{\max})$ and $\oplus \in \{+, -, \times, /\}$ and $\mathbf{u}$ is the unit roundoff then

$$\mathrm{fl}(x \oplus y) = (x \oplus y)(1+\delta), \text{ where } |\delta| \leq \mathbf{u}.$$

Loss of significance (or cancellation): If x and y are not floating-point numbers then we have to start with $\widehat{x} := \mathrm{fl}(x) = x(1+\delta_1)$ and $\widehat{y} := \mathrm{fl}(y) = y(1+\delta_2)$. Then

$$\widehat{x} \pm \widehat{y} = x(1+\delta_1) \pm y(1+\delta_2) = (x \pm y)\left(1 + \frac{x\delta_1 \pm y\delta_2}{x \pm y}\right)$$

Floating-point arithmetic

Arithmetic model: If $x, y \in F(\beta, t, e_{\min}, e_{\max})$ and $\oplus \in \{+, -, \times, /\}$ and $\mathbf{u}$ is the unit roundoff then

$$\mathrm{fl}(x \oplus y) = (x \oplus y)(1+\delta), \text{ where } |\delta| \leq \mathbf{u}.$$

Loss of significance (or cancellation): If x and y are not floating-point numbers then we have to start with $\widehat{x} := \mathrm{fl}(x) = x(1+\delta_1)$ and $\widehat{y} := \mathrm{fl}(y) = y(1+\delta_2)$. Then

$$\widehat{x} \pm \widehat{y} = x(1+\delta_1) \pm y(1+\delta_2) = (x \pm y)\left(1 + \frac{x\delta_1 \pm y\delta_2}{x \pm y}\right)$$

Now, if x and y have the same sign then the relative error $\left|\frac{x\delta_1 + y\delta_2}{x+y}\right| \leq |\delta_1| + |\delta_2| \leq 2\mathbf{u}$ is small but $\left|\frac{x\delta_1 - y\delta_2}{x-y}\right|$ can be arbitrarily large when $x - y$ is very small.

Example

Problem: Solve $ax^2 + bx + c = 0$, where $a, b, c \in \mathbb{R}$

Classical method: Naive use of the formula yields

$$x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

does not avoid subtraction whereas the modified formula

$$x_1 = -\frac{b + \mathrm{sign}(b)\sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{c}{ax_1}$$

avoid subtractions and hence avoid loss of significance.

Example

Problem: Solve $ax^2 + bx + c = 0$, where $a, b, c \in \mathbb{R}$

Classical method: Naive use of the formula yields

$$x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

does not avoid subtraction whereas the modified formula

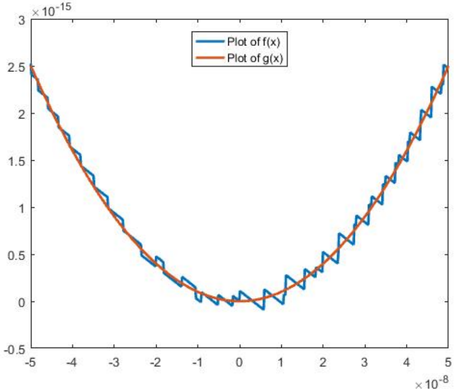
$$x_1 = -\frac{b + \mathrm{sign}(b)\sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{c}{ax_1}$$

avoid subtractions and hence avoid loss of significance.

For $10^{-3}x^2 + 10^7 x + 3 = 0$, the classical method in MATLAB yields $x_1 = -10^{10}$ and $x_2 = 0$ whereas the modified method yields the accurate answer $x_1 = -10^{10}$ and $x_2 = -3.0 \times 10^{-7}$.

Example

Evaluate $f(x) := e^x - \cos(x) - x$ for $-5 \times 10^{-8} \leq x \leq 5 \times 10^{-8}$.



Approximation of $f(x)$ using Taylor series avoids cancellation

$$f(x) = 1 + x + \frac{x^2}{2!} + \cdots - (1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \cdots) - x \approx x^2 + x^3/6 = g(x).$$

Cancellation and remedy

Let $f(x) := \dfrac{1-\cos(x)}{\sin(x)}$ for $x \neq 0$. If a is close to 0 then evaluation of $f(x)$ at a causes cancellation as $\cos(a) \approx 1$. The remedy is to rewrite $f(x)$ as

$$f(x) = \frac{1-\cos(x)}{\sin(x)} = \frac{\sin(x)}{1+\cos(x)}$$

which avoids cancellation at $a \approx 0$.

Rounding error of differentiation

Recall that $D_h f(x) := \dfrac{f(x+h) - f(x)}{h}$ approximates $f'(x)$. For small h, $D_h f(x)$ exhibits cancellation error due to rounding, which is magnified by the division of h.

Rounding error of differentiation

Recall that $D_h f(x) := \dfrac{f(x+h) - f(x)}{h}$ approximates $f'(x)$. For small h, $D_h f(x)$ exhibits cancellation error due to rounding, which is magnified by the division of h.

Let $\widehat{f(x)} := \mathrm{fl}(f(x))$ be the floating point approximation of $f(x)$. Then $\widehat{f(x)} = f(x)(1 + \delta(x))$, where $|\delta(x)| \leq \textbf{eps}$. Hence

Rounding error of differentiation

Recall that $D_h f(x) := \dfrac{f(x+h) - f(x)}{h}$ approximates $f'(x)$. For small h, $D_h f(x)$ exhibits cancellation error due to rounding, which is magnified by the division of h.

Let $\widehat{f(x)} := \mathrm{fl}(f(x))$ be the floating point approximation of $f(x)$. Then $\widehat{f(x)} = f(x)(1+\delta(x))$, where $|\delta(x)| \leq$ **eps**. Hence

$$\widehat{D_h f(x)} = \frac{\widehat{f(x+h)} - \widehat{f(x)}}{h} = \frac{f(x+h)(1+\delta(x+h)) - f(x)(1+\delta(x))}{h}$$

Rounding error of differentiation

Recall that $D_h f(x) := \dfrac{f(x+h) - f(x)}{h}$ approximates $f'(x)$. For small h, $D_h f(x)$ exhibits cancellation error due to rounding, which is magnified by the division of h.

Let $\widehat{f(x)} := \mathrm{fl}(f(x))$ be the floating point approximation of $f(x)$. Then $\widehat{f(x)} = f(x)(1+\delta(x))$, where $|\delta(x)| \leq$ **eps**. Hence

$$
\begin{aligned}
\widehat{D_h f(x)} &= \frac{\widehat{f(x+h)} - \widehat{f(x)}}{h} = \frac{f(x+h)(1+\delta(x+h)) - f(x)(1+\delta(x))}{h} \\
&= D_h f(x) + \frac{\delta(x+h)f(x+h) - \delta(x)f(x)}{h}.
\end{aligned}
$$

Rounding error of differentiation

Recall that $D_h f(x) := \dfrac{f(x+h) - f(x)}{h}$ approximates $f'(x)$. For small h, $D_h f(x)$ exhibits cancellation error due to rounding, which is magnified by the division of h.

Let $\widehat{f(x)} := \mathrm{fl}(f(x))$ be the floating point approximation of $f(x)$. Then $\widehat{f(x)} = f(x)(1+\delta(x))$, where $|\delta(x)| \leq \mathbf{eps}$. Hence

$$\begin{aligned}\widehat{D_h f(x)} &= \frac{\widehat{f(x+h)} - \widehat{f(x)}}{h} = \frac{f(x+h)(1+\delta(x+h)) - f(x)(1+\delta(x))}{h} \\ &= D_h f(x) + \frac{\delta(x+h)f(x+h) - \delta(x)f(x)}{h}.\end{aligned}$$

This shows that the rounding error

$$e_r(x,h) := \frac{|\delta(x+h)f(x+h) - \delta(x)f(x)|}{h} \leq \frac{\mathbf{eps}(|f(x+h)| + |f(x)|)}{h}.$$

Rounding error of differentiation

Recall that $E(x,h) := |f'(x) - D_h f(x)|$. By Taylor theorem

$$f(x+h) = f(x) + f'(x)h + h^2 f''(\xi)/2 \Longrightarrow E(x,h) = h|f''(\xi)|/2.$$

The best accuracy is obtained when $E(x,h) = e_r(x,h)$. Ignoring the constant terms gives

$$h \approx \mathbf{eps}/h \Longrightarrow h \approx \sqrt{\mathbf{eps}}$$

Rounding error of differentiation

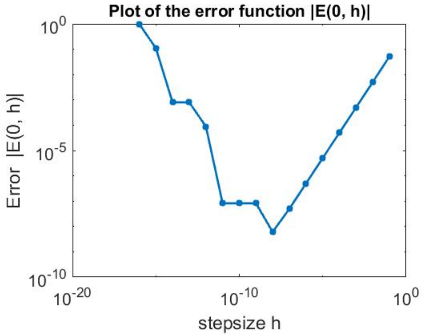
Recall that $E(x,h) := |f'(x) - D_h f(x)|$. By Taylor theorem

$$f(x+h) = f(x) + f'(x)h + h^2 f''(\xi)/2 \implies E(x,h) = h|f''(\xi)|/2.$$

The best accuracy is obtained when $E(x,h) = e_r(x,h)$. Ignoring the constant terms gives

$$h \approx \mathbf{eps}/h \implies h \approx \sqrt{\mathbf{eps}}.$$

For $f(x) = e^x$, we have



***