

Lecture 5

Finite Precision

L. Olson

Department of Computer Science
University of Illinois at Urbana-Champaign

2006.01.31



Review

floating point storage: 32-bit

$$\underbrace{\pm}_{1 \text{ bit}} \underbrace{0.b_1 b_2 \dots b_m}_{23 \text{ bits}} \times \underbrace{10^s}_{8 \text{ bits}}$$

Easy: Integer Example

represent $(25)_{10}$ in $(\cdot)_2$

$$\begin{array}{r} 12R1 \\ 6R0 \\ 25/2 \rightarrow 3R0 \rightarrow (110011)_2 \\ 1R1 \\ 0R1 \end{array}$$



Review

Moderate: Floating Point Example

represent $(25)_{10}$ in $(\cdot)_2$

| | k | 2^{-k} | b_k | r_k | |
|--|-----|----------|------------|--------|---------------------------|
| $3\frac{9}{16} \rightarrow 3.5625 \rightarrow$ | 0 | NA | $b_0 = NA$ | 0.5625 | $\rightarrow (11.1001)_2$ |
| | 1 | 0.5 | $b_1 = 1$ | 0.0625 | |
| | 2 | 0.25 | $b_2 = 0$ | 0.0625 | |
| | 3 | 0.125 | $b_3 = 0$ | 0.0625 | |
| | 4 | 0.0625 | $b_4 = 1$ | 0.0000 | |

Example

you try

- $(5.1)_{10} \rightarrow (?)_2$
- $(101.101)_2 \rightarrow (?)_{10}$

Review

Moderate: Floating Point Example

represent $(25)_{10}$ in $(\cdot)_2$

| | k | 2^{-k} | b_k | r_k | |
|--|-----|----------|------------|--------|---------------------------|
| $3\frac{9}{16} \rightarrow 3.5625 \rightarrow$ | 0 | NA | $b_0 = NA$ | 0.5625 | $\rightarrow (11.1001)_2$ |
| | 1 | 0.5 | $b_1 = 1$ | 0.0625 | |
| | 2 | 0.25 | $b_2 = 0$ | 0.0625 | |
| | 3 | 0.125 | $b_3 = 0$ | 0.0625 | |
| | 4 | 0.0625 | $b_4 = 1$ | 0.0000 | |

Example

you try

- $(5.1)_{10} \rightarrow (?)_2$
- $(101.101)_2 \rightarrow (?)_{10}$

Solution

Binary form of $x = 5.1$

| k | 2^{-k} | b_k | $r_k = r_{k-1} - b_k 2^{-k}$ |
|----------|--------------|-------|--|
| 0 | NA | NA | 0.1 |
| 1 | 0.5 | 0 | 0.1 |
| 2 | 0.25 | 0 | 0.1 |
| 3 | 0.125 | 0 | 0.1 |
| 4 | 0.0625 | 1 | $0.1 - 0.0625 = 0.0375$ |
| 5 | 0.03125 | 1 | $0.0375 - 0.03125 = 0.00625$ |
| 6 | 0.015625 | 0 | 0.00625 |
| 7 | 0.0078125 | 0 | 0.00625 |
| 8 | 0.00390625 | 1 | $0.00625 - 0.00390625 = 0.00234375$ |
| 9 | 0.001953125 | 1 | $0.00234375 - 0.001953125 = 0.000390625$ |
| 10 | 0.0009765625 | 0 | 0.000390625 |
| \vdots | \vdots | | |

Results in $(5.00011\ 0011 \dots)_2$

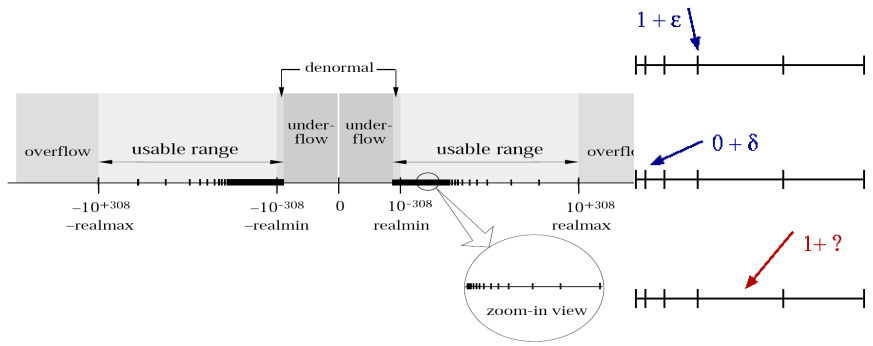


Round-off

two examples

- rounding a real to a float: $(0.2)_{10} = (0.0011\ 0011\ \dots)_2 \approx (0.0011)_2$
- arithmetic: $a + b = 1 + \epsilon + 0 + \delta \approx 1 + ?$

Floating Point Number Line



On Symbolic versus Numerical

Example

numerical

```
>> sin(pi)
```

ans =

```
1.2246e-016
```

Example

symbolic

```
>> pi=syms('pi')
```

```
>> sin(pi)
```

ans =

```
0
```



So What?

- we can only store so many digits
- this leads to rounding error in arithmetic as well

Example

Integer Arithmetic: can only store integers

$$1 + 10 = 4 \quad (\text{integer})$$

$$5 \times 4 = 20 \quad (\text{integer})$$

$$\frac{10}{2} = 5 \quad (\text{integer})$$

$$\frac{10}{3} = 3 \quad (\text{exact result is not an integer})$$

$$\frac{10}{300} = 0 \quad (\text{exact result is not an integer})$$

- we've made choices: round up, round down, round nearest?
- good project...



Floating point Arithmetic

Example

Only have 2 (decimal) mantissa

$$1.0 + 10.0 = 4 \quad (\text{FP exact})$$

$$5.0 \times 4.0 = 20.0 \quad (\text{FP exact})$$

$$\frac{10.0}{2.0} = 5 \quad (\text{FP exact})$$

$$\frac{10}{3} = 3.33 \quad (\text{FP is approximate})$$

$$\frac{10}{300} = 3.33 \times 10^{-2} \quad (\text{FP is approximate})$$



Where is the rounding error?

Example

example 1

```
>> format long e;  
>> x = 19/7  
x =  
    2.714285714285714e+000  
  
>> y = 7*x  
y =  
    19
```

Example

example 1

```
>> format long e;  
>> x = 29/1300  
x =  
    2.230769230769231e-02  
>> y = 29 - 1300*x  
y =  
    3.552713678800501e-015
```



Example: Quadratic Formula

The roots of $ax^2 + bx + c = 0$ are

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

For $a = 1.0$, $b = -54.32$, and $c = 0.1$, we have roots (11 digits)

$$x_1 = 54.3218158995 \quad x_2 = 0.0018410049576$$



Example: Quadratic Formula

Now suppose we have 4 digits. Do the arithmetic:

$$\begin{aligned}\hat{x} &= \frac{54.32 \pm \sqrt{(54.32)^2 - 0.400}}{2} \\ &= \frac{54.32 \pm \sqrt{2951 - 0.400}}{2} \\ &= \frac{54.32 \pm \sqrt{2951}}{2} \\ &= \frac{54.32 \pm \sqrt{54.32}}{2} \\ &= 0, 54.30\end{aligned}$$

error in 54.30

$$\frac{54.32 - 54.30}{54.32} \approx 0.04 = 4\% \text{ error}$$

error in 0

$$\frac{0.002 - 0.0}{0.002} \approx 1.00 = 100\% \text{ error}$$

A fix?

Perhaps the large numerator is throwing us off:

$$\begin{aligned}\tilde{x} &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \left(\frac{-b \mp \sqrt{b^2 - 4ac}}{-b \mp \sqrt{b^2 - 4ac}} \right) \\ &= \frac{2c}{-b \mp \sqrt{b^2 - 4ac}} \\ &= \frac{0.200}{54.32 \mp 54.32} \\ &= \frac{0.200}{108.6}, \frac{0.200}{0}\end{aligned}$$

!!!

catastrophic cancellation...not a good fix



Real (obscure) fix

We need to account for the sign of b :

Look at

$$q \equiv -\frac{1}{2} \left(b + \text{sign}(b) \sqrt{b^2 - 4ac} \right)$$

The definition of sign:

$$\text{sign}(b) = \begin{cases} 1 & \text{if } b \geq 0, \\ -1 & \text{else} \end{cases}$$

With this, the roots are given by

$$x_1 = \frac{q}{a} \quad x_2 = \frac{c}{q}$$

- HW!



Catastrophic Cancellation

Adding $c = a + b$ will result in a large error if

- $a \gg b$
- $a \ll b$

Let

$$a = x.xxx \cdots \times 10^0$$
$$b = y.yyy \cdots \times 10^{-8}$$

Then

$$\begin{array}{r} + \quad \overbrace{0.000\ 0000\ yyy\ yyy}^{\text{finite precision}} \quad yyy\ yyy \\ = \quad x.xxx\ xxxx\ zzzz\ zzzz \quad \underbrace{????\ ???}_{\text{lost precision}} \end{array}$$



Catastrophic Cancellation

Subtracting $c = a - b$ will result in large error if $a \approx b$. For example

$$\begin{array}{r} ax.xxxx\ xxxx\ xxx1 \overbrace{ssss\dots}^{\text{lost}} \\ bx.xxxx\ xxxx\ xxx0 \overbrace{tttt\dots}^{\text{lost}} \end{array}$$

Then

$$\begin{array}{r} \overbrace{x.xxx\ xxxx\ xxx1}^{\text{finite precision}} \\ + \overbrace{x.xxx\ xxxx\ xxx0}^{\text{finite precision}} \\ \hline = 0.000\ 0000\ 0001 \underbrace{????\ ????}_{\text{lost precision}} \end{array}$$



Summary

- addition: $c = a + b$ if $a \gg b$ or $a \ll b$
- subtraction: $c = a - b$ if $a \approx b$
- catastrophic: caused by a single operation, not by an accumulation of errors
- can often be fixed by mathematical rearrangement



How large is the roundoff error?

- We measure this by relating to the *machine precision*, ϵ .
- Viewing the floating number line, there is a number ϵ so that if $\delta < \epsilon$

$$1 + \delta = 1$$

```
eps
>> help eps
EPS Spacing of floating point numbers.
    D = EPS(X), is the positive distance from ABS(X)
    to the next larger in magnitude floating point
    number of the same precision as X. X may be
    either double precision or single precision.
        .
        .
        .
>> eps(1)

ans =
    2.220446049250313e-016

>> 2^(-52)

ans =
    2.220446049250313e-016
```



To compute ϵ

```
epsilon = 1;
it = 0;
maxit = 10000;
while it < maxit
    epsilon = epsilon/2;
    b = 1 + epsilon;
    if b==1
        break;
    end
    it=it+1;
end
```

Practical Considerations

- when comparing floating point numbers, do not use exact equivalence (for example ==)
- terminate iterations when *epsilon* machine precision is reached

bad

```
if (x==y)
  .
  .
  .
end
```

good

```
if (abs(x-y) < tol)
  .
  .
  .
end
```

How close is float x to float y ?

We use two measures of error:

- absolute error
- relative error

How close does \hat{x} approximate an exact value x ?

Absolute Error $error_{abs} = |\hat{x} - x|$

Relative Error $error_{rel} = \frac{|\hat{x} - x|}{|x|}$



Why Relative?

Example

For small x ,

$$\sin(x) = x - \frac{x^3}{3!} + \dots \approx x$$

Then

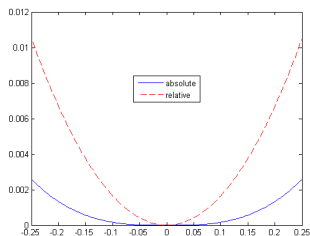
$$error_{abs} = |x - \sin(x)| = \frac{x^3}{3!} - \frac{x^5}{5!} + \dots$$

$$error_{rel} = \frac{|x - \sin(x)|}{|\sin(x)|} = \frac{x}{\sin(x)} - 1$$



Relative vs. Absolute

```
>> x=linspace(-0.25,0.25,200);  
>> eabs = abs(sin(x) - x);  
>> erel = x./sin(x) - 1;  
>> plot(x,eabs,'b-',x,erel,'r--');  
>> legend('absolute','relative')
```



Truncation Error

The Taylor series expansion of $\sin(x)$ is

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$$

If $x \ll 1$, then the remaining terms are small. If we neglect these terms

$$\sin(x) = \underbrace{x - \frac{x^3}{3!} + \frac{x^5}{5!}}_{\text{approximation to sin}} - \underbrace{\frac{x^7}{7!} + \frac{x^9}{9!} - \dots}_{\text{truncation error}}$$

sinetrunc.m

Taylor Series revisited

For a sufficiently smooth function $f(x)$ defined on $[a, b]$, the n^{th} order Taylor Series approximation is

$$\begin{aligned}f_n(x) &= f(x_0) + (x - x_0) \frac{df}{dx} \Big|_{x=x_0} \\ &\quad + \frac{(x - x_0)^2}{2} \frac{d^2f}{dx^2} \Big|_{x=x_0} \\ &\quad + \dots + \frac{(x - x_0)^n}{n!} \frac{d^n f}{dx^n} \Big|_{x=x_0}\end{aligned}$$

In fact, $f(x)$ can be written exactly as a finite sum for some $\xi \in [x_0, x]$:

$$\begin{aligned}f(x) &= f(x_0) + (x - x_0) \frac{df}{dx} \Big|_{x=x_0} \\ &\quad + \frac{(x - x_0)^2}{2} \frac{d^2f}{dx^2} \Big|_{x=x_0} \\ &\quad + \frac{(x - x_0)^n}{n!} \frac{d^n f}{dx^n} \Big|_{x=x_0} \\ &\quad + \dots + \frac{(x - x_0)^{n+1}}{(n+1)!} \frac{d^{n+1} f}{dx^{n+1}} \Big|_{x=\xi}\end{aligned}$$



Taylor Series revisited

So the function $f(x)$ can be written as the Taylor Series approximation plus an error (truncation) term:

$$f(x) = f_n(x) + R_n(x)$$

where

$$R_n(x) = \frac{(x - x_0)^{n+1}}{(n + 1)!} \left. \frac{d^{n+1}f}{dx^{n+1}} \right|_{x=\xi}$$



Big "O"

We write the error as

$$\begin{aligned} R_n(x) &= \frac{(x - x_0)^{n+1}}{(n + 1)!} \left. \frac{d^{n+1} f}{dx^{n+1}} \right|_{x=\xi} \\ &= \mathcal{O}\left(\frac{(x - x_0)^{n+1}}{(n + 1)!}\right) \end{aligned}$$

since we assume the $(n + 1)^{\text{th}}$ derivative is bounded on the interval $[a, b]$.

Often, we let $h = x - x_0$ and we have

$$f(x) = f_n(x) + \mathcal{O}(h^{n+1})$$



Example

Let

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

The Taylor Series approximation of order 0, 1, and 2 are

$$f_0 = \frac{1}{1-x_0}$$

$$f_1 = \frac{1}{1-x_0} + \frac{x-x_0}{(1-x_0)^2}$$

$$f_2 = \frac{1}{1-x_0} + \frac{x-x_0}{(1-x_0)^2} + \frac{(x-x_0)^2}{(1-x_0)^3}$$



Example

