553.385 Numerical Linear Algebra, Spring 2022
Section 2

Hongyu Cheng

hongyucheng@jhu.edu

February 21, 2022

1 Preliminaries

Limit of a sequence: We say that a sequence x_n converges if there exists $L \in \mathbb{R}$ such that $\forall \varepsilon > 0, \exists N \in \mathbb{N}_+$ such that $\forall n > N$ we have $|x_n - L| < \varepsilon$.
Example: Problem 1 in Homework 1

Intermediate Value Theorem: If $f \in C[a, b]$ and M is any number between $f(a)$ and $f(b)$, then $\exists c \in (a, b)$ such that $f(c) = M$.

Taylor's Theorem: Suppose $f \in C^n[a, b]$, $f^{(n+1)}$ exists on $[a, b], x_0 \in [a, b]$. For each $x \in [a, b]$ there exists $\xi(x)$ between x_0 and x such that

$$f(x) = \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)^{n+1}$$

Example: Problem 2 in Homework 1

IEEE Standard 754 Floating Point Numbers:

$\hat{x} = \sigma M \cdot 2^E$, where $M = 1 + F = (1.a_1 \cdots a_p)_2 \in [1, 2), E \in [L, U)$

IEEE SINGLE: $\underbrace{1 \text{ bit}}_{\text{sign}} + \underbrace{8 \text{ bits}}_{\text{exponents}} + \underbrace{23 \text{ bits}}_{\text{fraction}} = 32 \text{ bits}$

IEEE DOUBLE: $\underbrace{1 \text{ bit}}_{\text{sign}} + \underbrace{11 \text{ bits}}_{\text{exponents}} + \underbrace{52 \text{ bits}}_{\text{fraction}} = 64 \text{ bits}$

Example: Problem 4 in homework 1, `x=3ff6a0902de00d1b`

2 Examples

Example 1(*Computer Arithmetic*). Given a number x, consider the following questions:

1. What is the representation of the number x in base-2? Write the number in the form $x = (1 + F) \times 2^E$ where fraction $F \in [0, 1)$ should be given its base-2 representation and the exponent E is an integer.
2. Is x a machine number in IEEE standard single precision arithmetic?

(i) $x = 1/8$.

(ii) $x = 1/5$.

(iii) $x = 2^{128} - 2^{104}$

Example 2(*Source of error*). Consider the symmetric-difference approximation to the second-derivative

$$D_h^2 f(x) = \frac{f(x+h) + f(x-h) - 2f(x)}{h^2} \approx f''(x),\ h > 0 \tag{1}$$

Use the Taylor expansion and intermediate value theorem, we can prove that for any 4 time continuously differentiable function f

$$D_h^2 f(x) = f''(x) + \frac{1}{12} f^{(4)}(\xi) h^2,\ \xi \in [x-h, x+h] \tag{2}$$

Thus, when $f(x) = \frac{1}{6}x^3$, according to (2), there is no error in $D_h^2 f(x)$ for any $h > 0$, in which case one has

$$D_h^2 f(x) = x = f''(x) \text{ for all } h > 0$$

in exact arithmetic. Evaluate the approximation $D_h^2 f(x)$ numerically in MATLAB with single precision arithmetic for $f(x) = \frac{1}{6}x^3$ with $x = 2$ for the values $h = 10^{1-i/2}, i = 1, 2, 3, \ldots, 11$. Give the relative error for each choice of h.

We can see that only the first three approximations for the largest h are accurate to approximately single-precision, but the errors become worse as h gets smaller. This is a loss of significance error due to evaluating in finite-precision arithmetic a sum of three terms which is close to zero in exact arithmetic.

When $f(x) = \frac{1}{8}x^4$, use equation (2) to obtain an optimal h_* for which the error is minimum in single precision arithmetic.

Truncation error dominating to the right and round-off error to the left.

Example 3(*Error propagation*). The following infinite series can be exactly evaluated as

$$S := \sum_{n=1}^{\infty} \frac{2n+1}{n^2(n+1)^2} = 1. \tag{3}$$

Further, the k-th partial sum S_k and its remainder $R_k = 1 - S_k$ are given by

$$S_k = \sum_{n=1}^{k} \frac{2n+1}{n^2(n+1)^2} = 1 - \frac{1}{(k+1)^2}, \text{ where } R_k = \frac{1}{(k+1)^2}. \tag{4}$$

Then consider the following MATLAB script to evaluate approximately the infinite series:

```
nn = 1;
dS = (2*nn+1)/nn^2/(nn+1)^2;
Sold = 0;
S = dS;
while (abs(S-Sold)>0)
    nn = nn+1;
    dS = (2*nn+1)/nn^2/(nn+1)^2;
    Sold = S;
    S = S+dS;
end
```

:

Double precision was not obtained in the above algorithm. This occurred because the sum was terminated for $n = 330281$ when the remainder was 10,000 times larger than 10^{-16}. This happened because the next term $\frac{2n+1}{n^2(n+1)^2}$ in the sum with $n = 330282$ is only 5.5×10^{-17}, which is smaller than the unit round in MATLAB. Hence, adding this tiny number to $S_{330281} \approx 1$ did not lead to an improved value of S_{330282} so that the while-loop terminated. To avoid this problem, one should always sum numbers in floating-point arithmetic from small terms to large ones, instead of from large to small. To decide how large an n must be considered, we use the remainder to find an n so than $R_n = 1/(n+1)^2 \approx eps$, and thus we find that $n = 6710884 \approx 1/\sqrt{eps}$.

```
    N = round(1/sqrt(eps))
    n = N;
    R = 1/(n+1)^2
    dS = (2*n+1)/n^2/(n+1)^2;
    S = dS;
    for n = N-1:-1:1
        dS = (2*n+1)/n^2/(n+1)^2;
        S = S+dS;
    end
    S = S
    relerr=abs(S-1)
```

:

$$D_h^2 f(x) = \frac{f(x+h) + f(x-h) - 2f(x)}{h^2} \approx f''(x), \ h > 0 \tag{5}$$