Lecture January 12:

General Atomics, UAVs and floating point arithmetic?

solution to last practice problem

$$f(x) = \sqrt{x} \quad f'(x) = (1/2)x^{-1/2} \quad f''(x) = -(1/4)x^{-3/2} \quad f'''(x) = (3/8)x^{-5/2} \tag{1}$$
$$x_0 = 1 \tag{2}$$
$$P_2(x) = 1 + (1/2)(x-1) - (1/4)(x-1)^2/2 = 1 + (x-1)/2 - (x-1)^2/8 \tag{3}$$
$$P_2(2) = 1 + 1/2 - 1/8 = 11/8 \tag{4}$$
$$|\sqrt{2} - 11/8| = 0.039 \tag{5}$$

error bound:

$$R_2(x) = (3/8)\xi(x)^{-5/2}(x-1)^3/6 \tag{6}$$
$$\max_{1\le x\le 2} |R_2(x)| \le \max_{1\le x\le 2} |(3/8)x^{-5/2}| \max_{1\le x\le 2} |(x-1)^3/6| = \tag{7}$$
$$(3/8)(1/6) = 1/16 \tag{8}$$

base 2 arithmetic vs. base 10 arithmetic example: 43=32+8+2+1, 43=101011=1.01011x2^5
example: 15.625=8+4+2+1+1/2+1/8=1111.101=1.111101x2^3
example: 0.1 ? 0.1-1/16=0.0375, 0.1-1/16-1/32=0.00625, 0.1-1/16-1/32-1/256=0.00234375, 0.1-1/16-1/32-1/256-1/512=0.000390625, . . . , 0.1=0.000110011 . . . =1.10011. . . x 2^{-4}

floating point representation double precision: $(-1)^s 2^{c-1023}(1+f)$ (1+11+52 bits)
single precision: $(-1)^s 2^{c-127}(1+f)$ (1+8+23 bits)
for 43, $c = 132 = 128 + 4$ and $f = 0.01011$,

$$43 = 0 \quad 10000100 \quad 01011000000000000000000$$

15.625, $c = 130 = 128 + 2$ and $f = 0.111101$

$$15.625 = 0 \quad 10000010 \quad 11110100000000000000000$$

next larger representable number example,

$$\begin{gathered} 0 \quad 10000100 \quad 01011000000000000000000\,+ \\ 0 \quad 10000100 \quad 00000000000000000000001 = \\ 0 \quad 10000100 \quad 01011000000000000000001 = \\ 43 + 2^5 2^{-23} \approx 43 + 3.8 \times 10^{-6} \end{gathered}$$

next smaller representable number example,

$$\begin{gathered} 0 \quad 10000100 \quad 01011000000000000000000\,- \\ 0 \quad 10000100 \quad 00000000000000000000001 = \\ 0 \quad 10000100 \quad 01010111111111111111111 = \\ 43 - 2^5 2^{-23} \approx 43 - 3.8 \times 10^{-6} \end{gathered}$$

Zero representation: 0 00000000 00000000000000000000000

smallest magnitude number:

$$0 \quad 00000001 \quad 00000000000000000000000 = 2^{-126} \approx 1/(8.5 \times 10^{37})$$

largest magnitude number:

$$0 \quad 11111110 \quad 11111111111111111111111 = 2^{127}(2 - 2^{-23}) \approx 1.7 \times 10^{38}$$

k **digit chopping**

$$y = 0.d_1 d_2 \dots d_k d_{k+1} \dots \times 10^n$$
$$fl(y) = 0.d_1 d_2 \dots d_k \times 10^n$$

examples: $k = 3$ and $y = 10.000323$, $y = 0.0001217$, $y = 1021 \times 10^{23}$

k **digit rounding**

$$fl(y) = \text{chop}(y + 5 \times 10^{n-(k+1)})$$

examples: $k = 3$ and $y = 10.000323$, $y = 0.0001217$, $y = 1021 \times 10^{23}$

absolute vs. relative error absolute error: $|p - p^*|$ where p^* is approximation.
relative error: $|p - p^*|/|p|$ where p^* is approximation.

significant digits t

$$|p - p^*|/|p| \leq 5 \times 10^{-t}$$

For k digit chopping,

$$\left|\frac{y - fl(y)}{y}\right| \leq 10^{-k+1}$$

loss of precision

$$fl(x) = 0.d_1 \dots d_p \alpha_{p+1} \dots \alpha_k \times 10^n$$
$$fl(y) = 0.d_1 \dots d_p \beta_{p+1} \dots \beta_k \times 10^n$$
$$fl(fl(x) - fl(y)) = 0.\sigma_{p+1} \dots \sigma_k \times 10^{n-p}$$
$$0.\sigma_{p+1} \dots \sigma_k = 0.\alpha_{p+1} \dots \alpha_k - 0.\beta_{p+1} \dots \beta_k$$

Loss of precision occurs when approximating the derivative of a function:

$$\frac{f(x+h) - f(x)}{h}$$

h is small.

In class group work Convert the following numbers into single and double precision format:

$$52, 19.5, 3.1$$

for 3.1, not necessary to expand all the binary digits, just enough so that I see you understand.