CMPS 251
Assignment 2-Solution

Problem 1

Code

```
function y=dec2binfull(x)
if (x>32767)
    error('The input number is larger than the maximum allowed');
end
y=zeros(1,30);
integer=floor(x);
decimal=x-floor(x);
count=15;
while integer >= 1
    y(count)=mod(integer,2);
    integer=floor(integer/2);
    count=count-1;
end
count=16;
while sum(decimal~=[0 1])>1 && count<=30
    decimal=decimal*2;
    y(count)=floor(decimal);
    decimal=decimal-y(count);
    count=count+1;
end
% to represent as a string
y=[num2str(y(1:15)) '.' num2str(y(16:30))];
```

Example

```
>> dec2binfull(52.234375)

ans =

0  0  0  0  0  0  0  0  0  1  1  0  1  0  0.0  0  1  1  1  1  0  0  0  0  0  0  0  0  0
```

Problem 2

Part 1

a) Using six significant digits, cos(0.007)=0.999976 and sin(0.007)=0.00699994. Thus, f(0.007)=(1-0.999976)/0.00699994= 0.00342860.

b) Using Matlab, the value is as follows

```
>> format long
>> f=(1-cos(0.007))/sin(0.007)
f =
0.00350001429173
```

The relative error is then

$$\text{error} = \frac{|0.00350001429173 - \ 0.00342860|}{0.00350001429173} = 0.0204$$

c) Computing the function f near 0 causes a problem due to the cancellation error that occurs when subtracting two nearly equal numbers (cos(0.0...) and 1). To rearrange the function f, we multiply the numerator and denominator by 1+cos(x) as follows

$$f(x) = \frac{1-\cos(x)}{\sin(x)} \cdot \frac{1+\cos(x)}{1+\cos(x)} = \frac{1-\cos^2(x)}{\sin(x)+\sin(x)\cos(x)} = \frac{\sin^2(x)}{\sin(x)\,(1+\cos(x))}$$
$$= \frac{\sin(x)}{1+\cos(x)}$$

Using the values of sin(0.007) and cos(0.007) in part (a) with the reformulated function yields f(0.007)=0.00350001 for a relative error of 1.22×10^{-6}

Part 2

The function f clearly causes overflow for large values of x and y. The technique to follow while dealing with such problems is to bound the result for large values of x and y. One way to reformulate the function is as follows

$$f(x,y) = \sqrt{x^2+y^2} = \begin{cases} |y|\sqrt{1+\left(\frac{x}{y}\right)^2} & \text{if } 0 \le |x| \le |y| \\ |x|\sqrt{1+\left(\frac{y}{x}\right)^2} & \text{if } 0 \le |y| \le |x| \end{cases}$$

Depending on the values of x and y, the selected ratio is less than 1 and the magnitude of the results is close to that of |x| or |y|. The remaining issues would be when x and y cannot be represented using the machine floating point representation.

<u>Problem 3</u>

The derivation for the error in the FunkyRound scheme uses the same approach that was employed in class. The calculations are shown for the absolute error. The relative error can be calculated using the minimum bound on x

1-For $d_{t+1}<\beta/2$

For this case, and in contrast to the normal rounding, we are rounding up. The maximum error occurs when all the digits $d_i, i = t+1, \cdots, \infty$ have a value of zero

$$|x - \text{fl}(x)| < \left| \sum_{i=t+1}^{\infty} d_i \beta^{-i} - \beta^{-t} \right| \cdot \beta^e = \beta^{-t} \cdot \beta^e$$

2-For $d_{t+1}>\beta/2$

For this case, rounding down is done. The maximum error occurs when all remaining digits take the highest value which is β-1

Hence, the absolute error is given as

$$|x - \mathrm{fl}(x)| < (\beta - 1)\sum_{i=t+1}^{\infty} \beta^{-i} \cdot \beta^{e} = (\beta - 1)\left(\frac{1}{1 - \frac{1}{\beta}} - \frac{1 - \left(\frac{1}{\beta}\right)^{t+1}}{1 - \frac{1}{\beta}}\right) \cdot \beta^{e} = \beta^{-t} \cdot \beta^{e}$$

3- For $d_{t+1}=\beta/2$

For this case, we round by setting d_t to 0 and removing the remaining decimal digits. The maximum error in this case occurs when $d_t=\beta-1$, $d_{t+1}=\beta/2$ and the remaining digits take the maximum value of β-1. The error is then given as

$$|x - \mathrm{fl}(x)| < \left((\beta - 1) \cdot \beta^{-t} + \frac{\beta}{2} \cdot \beta^{-t-1} + (\beta - 1)\sum_{i=t+2}^{\infty} \beta^{-i}\right) \cdot \beta^{e}$$

$$= \left((\beta - 1) \cdot \beta^{-t} + \frac{1}{2} \cdot \beta^{-t} + \beta^{-t-1}\right) \cdot \beta^{e}$$

$$= ((\beta - 1/2) \cdot \beta^{-t} + \beta^{-t-1}) \cdot \beta^{e}$$

$$= \left(\beta - \frac{1}{2} + \frac{1}{\beta}\right)\beta^{-t} \cdot \beta^{e}$$

$$\frac{2\beta^2 - \beta + 2}{2\beta}\beta^{-t} \cdot \beta^{e}$$

The factor $\frac{2\beta^2-\beta+2}{2\beta}$ is bigger than 1 for any base, hence the last case yields the largest error for this fictitious rounding scheme.

Problem 4

Part 1

1- +0

2- -0

3- +∞

4- -∞

5- $(1.0)_2 . 2^{1-127}=2^{-126}=1.1755 \times 10^{-38}$

6- $(1.011)_2 . 2^{129-127}=(101.1)_2=5.5$

7- $(1.0)_2 . 2^{127-127}=1$

8- $(1.10011001100110011001100)_2 . 2^{123-127}=(0.00011001100110011001100)_2$
$=0.099999994039536 \approx 0.1$

Part 2

[C705A700]

Binary: 1 10001110 00001011010011100000000

Decimal: -34215

[45607000]
Binary: 0 10001010 11000000111000000000000
Decimal: 3591

[494F96A0]
Binary: 0 10010010 10011111001011010100000
Decimal: 850282
Calculations:
Sign bit 0 → +
Exponent: $(10010010)_2$=146 → exponent =$2^{146-127}$ =2^{19}.
1.f: $(1.10011111001011010100000)_2$
Number: $(1.10011111001011010100000)_2$ x 2^{19}=$(011\ 001\ 111\ 100\ 101\ 101\ 010)_2$=$(3174552)_8$ =850282

Part 3
64.37109375
Binary: 1000000.01011111=$(1.00000001011111) \times 2^6$
Exponent: c-127=6 → c=133 = $(10000101)_2$
Sign bit: 0
Single precision representation: 0 10000101 00000001011111000000000
$[4280BE00]_{16}$

0.234375
Binary: 0.001111 =1.111×2^{-3}
Exponent: c-127=-3 → c=124 = $(0111\ 1100)_2$
Single precision representation: 0 01111100 11100000000000000000000
$[3E700000]_{16}$

-9876.54321
Binary: 10011010010100.10001011000011111100111 (non-terminating)
=$(1.00110100101001000101100)_2 \times 2^{13}$ since next bit is 0
Exponent: c=140 =$(10001100)_2$
Sign bit : 1
Single precision representation: 1 10001100 00110100101001000101100
$[C61A522C]_{16}$
Since the value is not exact, we calculate the relative error as

$$\frac{9876.54321 - 9876.54296875}{9876.54321} = 2.44 \times 10^{-8}$$

Problem 5

fl(fl(a+b)+c)

a = $0.23371258\times 10^{-4} = 0.00000023371258\times 10^{2}$

b=0.33678429×10^{2}

a+b=$0.33678452371258\times 10^{2}$

Since we have an 8-decimal digit precision

fl(a+b)=0.33678452×10^{2}

Now

fl(fl(a+b)+c)=fl($0.33678452\times 10^{2} - 0.33677811 \times 10^{2}$)

= fl(0.00000641×10^{2}))

=0.64100000×10^{-3}

fl(a+fl(b+c))

fl(b+c)=fl($0.33678429\times 10^{2} - 0.33677811 \times 10^{2}$)

=fl(0.00000618×10^{2})

=0.61800000×10^{-3}

a = $0.23371258\times 10^{-4} = 0.023371258\times 10^{-3}$

fl(a+fl(b+c))=fl($0.61800000\times 10^{-3} + 0.023371258\times 10^{-3}$)

=fl($0.641371258\times 10^{-3}$) = 0.64137126×10^{-3}

Looking at the two results we can see that the second operation has 8 significant digits and a precision equal to the machine precision. The first operation resulted in an accuracy of 3 significant digits (not counting the spurious zeros) and thus a loss of significance of 5 digits. This is due to the cancellation error in adding the sum of a and b to c.

Now we consider machine number a,b, and c and neglecting higher order terms in δ

$$\mathrm{fl}(\mathrm{fl}(a+b)+c) = \big((a+b)(1+\delta_1)+c\big)(1+\delta_2) \approx a+b+c+(a+b)\delta_1+(a+b+c)\delta_2$$

$$\mathrm{fl}(a+\mathrm{fl}(b+c)) = \big((b+c)(1+\delta_3)+a\big)(1+\delta_4) \approx a+b+c+(b+c)\delta_3+(a+b+c)\delta_4$$

Now finding the relative error for both cases

$$\left|\frac{a+b+c-(a+b+c+(a+b)\delta_1+(a+b+c)\delta_2)}{a+b+c}\right| \le \left|\frac{a+b}{a+b+c}\epsilon+\epsilon\right|$$

$$= \left|\frac{a+b}{a+b+c}+1\right|\epsilon$$

$$\left|\frac{a+b+c-(a+b+c+(b+c)\delta_3+(a+b+c)\delta_4)}{a+b+c}\right| \le \left|\frac{b+c}{a+b+c}\epsilon+\epsilon\right|$$

$$= \left|\frac{b+c}{a+b+c}+1\right|\epsilon$$

Finally, we can note that when $|b + c| < |a + b|$, the second operation is better than the first one.