Tutorial 2 Solution

25 January 2023 11:19 AM

1. Let X be a sufficiently large number which result in an overflow of memory on a computing device. Let x be a sufficiently small number which result in underflow of memory on the same computing device. Then give the output of the following operations:

(i) $x \times X$ (ii) $3 \times X$ (iii) $3 \times x$ (iv) x/X (v) X/x.

Ans : (i) NaN (ii) inf (iii) 0 (iv) 0 (v) inf

2. In this question, computations are done on a computer which uses 3-digit chopping arithmetic.

   i) Compute the mid-point of the interval $[0.982, 0.987]$. Show all the steps of the computation.

   ii) Give a different way of computing the mid-point of $[0.982, 0.987]$, so that the mid-point lies in the given interval $[0.982, 0.987]$. Show all the steps of the computation. Explain the differences between both these computations.

i)

Midpoint = (Lower endpoint + Upper endpoint)/2

This way of calculating the mid-point guarantees that the mid-point will always lie within the given interval. This is because it is the average of the lower and upper endpoint of the interval.

Step 1: Find the lower endpoint: 0.982
Step 2: Find the upper endpoint: 0.987
Step 3: Add the lower and upper endpoint: 0.982 + 0.987 = 1.969
Step 4: Divide the sum by 2: 1.969/2 = 0.9845

So, the midpoint of the interval [0.982, 0.987] is 0.9845

The difference between both these computations is that in the first method, the rounding up of the computation is done after the final result, while in this method, the guarantee that the mid-point will always lie within the given interval is provided by the formula itself which is the average of the lower and upper endpoint.

ii)

Another way to compute the midpoint of the interval [0.982, 0.987], so that the mid-point lies in the interval, is to use the following formula:

Midpoint = Lower endpoint + (Upper endpoint - Lower endpoint)/2

This way of calculating the mid-point guarantees that the mid-point will always lie within the given interval.

Step 1: Find the lower endpoint: 0.982
Step 2: Find the upper endpoint: 0.987
Step 3: Subtract the lower endpoint from the upper endpoint: 0.987 - 0.982 = 0.005
Step 4: Divide the result by 2: 0.005/2 = 0.0025
Step 5: Add the result to the lower endpoint: 0.982 + 0.0025 = 0.9845

So, the midpoint of the interval [0.982, 0.987] is 0.9845

The difference between this method and the previous methods is that this method subtracts the lower endpoint from the upper endpoint and then add half of that difference to the lower endpoint, which guarantees that the mid-point will always lie within the given interval, regardless of any rounding or chopping errors that may occur.

3.

In the following problems, show all the steps involved in the computation.

i) Using 5-digit rounding, compute $37654 + 25.874 - 37679$.

ii) Let $a = 0.00456$, $b = 0.123$, $c = -0.128$. Using 3-digit rounding, compute $(a + b) + c$, and $a + (b + c)$. What is your conclusion?

iii) Let $a = 2$, $b = -0.6$, $c = 0.602$. Using 3-digit rounding, compute $a \times (b+c)$, and $(a \times b) + (a \times c)$. What is your conclusion?

i)

To compute 37654 + 25.874 − 37679 using 5-digit rounding, we will round each number to 5 decimal places before performing the arithmetic operation.

Step 1: Round 37654 to 5 decimal places: 37654

Step 2: Round 25.874 to 5 decimal places: 25.874

Step 3: Round 37679 to 5 decimal places: 37679

Step 4: Perform the arithmetic operation: 37654 + 25.874 − 37679 = 37654 + 25.874 - 37679 = 37654 + 25.874 - 37679 = -24.126

Since the computation is done on a computer which uses 5-digit rounding arithmetic, the final answer will be rounded to 5 decimal places.

So, the result of the computation 37654 + 25.874 − 37679 using 5-digit rounding is -24.126

Ii)

To compute (a + b) + c and a + (b + c) using 3-digit rounding, we will round each number to 3 decimal places before performing the arithmetic operation.

Step 1: Round 0.00456 to 3 decimal places: 0.00456

Step 2: Round 0.123 to 3 decimal places: 0.123

Step 3: Round -0.128 to 3 decimal places: -0.128

Step 4: Perform the arithmetic operation (a + b) + c = (0.00456 + 0.123) + (-0.128) = 0.12756 + (-0.128) = -0.00084

Step 5: Perform the arithmetic operation a + (b + c) = 0.00456 + (0.123 + (-0.128)) = 0.00456 + (-0.005) = 0.00056

We can see that the order of the operation does matter when the numbers are rounded to the same number of decimal places. The result of (a+b)+c is not equal to the result of a+(b+c)

So, the result of (a + b) + c using 3-digit rounding is -0.00084 and the result of a + (b + c) using 3-digit rounding is 0.00056.

Therefore, the conclusion is that when the numbers are rounded to the same number of decimal places, the order of operations do matter, and it can lead to different results.

Iv)

To compute a×(b+c) and (a × b)+(a × c) using 3-digit rounding, we will round each number to 3 decimal places before performing the arithmetic operation.

Step 1: Round 2 to 3 decimal places: 2

Step 2: Round -0.6 to 3 decimal places: -0.6

Step 3: Round 0.602 to 3 decimal places: 0.602

Step 4: Perform the arithmetic operation a×(b+c) = 2×(-0.6 + 0.602) = 2×(-0.002) = -0.004

Step 5: Perform the arithmetic operation (a × b)+(a × c) = (2 × -0.6) + (2 × 0.602) = -1.2 + 1.204 = 0.004

Step 5: Perform the arithmetic operation (a × b)+(a × c) = (2 × -0.6) + (2 × 0.602) = -1.2 + 1.204 = 0.004

We can see that the order of the operation does not matter in this case. The result of a×(b+c) is equal to the result of (a × b)+(a × c).

So, the result of a×(b+c) using 3-digit rounding is -0.004 and the result of (a × b)+(a × c) using 3-digit rounding is 0.004

Therefore, the conclusion is that when the numbers are rounded to the same number of decimal places, the order of operations does not change the result and it is always the same.

This Solution may be wrong so don't completely relay on this.

4.

Consider a computing device having exponents e in the range $m \leq e \leq M$, $m, M \in \mathbb{Z}$. Let n be an integer such that $n \leq |m| + 1$.

i) If the device uses n-digit rounding binary floating-point arithmetic, then show that $\delta = 2^{-n}$ is the machine epsilon.

ii) What is the machine epsilon of the device if it uses n-digit rounding decimal floating-point arithmetic? Justify your answer.

i)

In a computing device that uses n-digit rounding binary floating-point arithmetic, the machine epsilon (δ) is defined as the smallest positive value such that 1 + δ ≠ 1. In other words, it is the smallest value that can be added to 1 and still be considered different from 1 by the device.

In this case, the device uses exponents in the range m ≤ e ≤ M and n is an integer such that n ≤ |m| + 1. Since the device uses binary floating-point arithmetic, the smallest representable number is 2^m and the largest representable number is 2^M.

Now, the machine epsilon can be found by considering the next representable number after 1. The next representable number after 1 is 1 + 2^(1-n) . The smallest representable value greater than 1 is 1 + 2^(1-n), and that is the machine epsilon.

So, the machine epsilon for this device is δ = 2^(1-n) = 2^(-n) = 2^-n

Therefore, if the device uses n-digit rounding binary floating-point arithmetic, then the machine epsilon is δ = 2^(-n)

ii)

In a computing device that uses n-digit rounding decimal floating-point arithmetic, the machine epsilon (δ) is defined as the smallest positive value such that 1 + δ ≠ 1. In other words, it is the smallest value that can be added to 1 and still be considered different from 1 by the device.

In this case, the device uses n-digit rounding decimal floating-point arithmetic.

For decimal floating-point arithmetic, the machine epsilon is given by 10^(-n), where n is the number of decimal digits used in the representation of the numbers. This is because, in decimal arithmetic, the least significant digit is the one on the right of the decimal point and the least significant digit is the one that will be rounded off when n-digit rounding is used.

So, the machine epsilon for this device is δ = 10^(-n)

Therefore, if the device uses n-digit rounding decimal floating-point arithmetic, then the machine epsilon is δ = 10^(-n)

5.

Consider a computing device that uses n-digit chopping (decimal) arithmetic. Let $\mathrm{fl}(x)$ denote the floating-point approximation of a positive real number x in this device. Prove

$$\left|\frac{x - \mathrm{fl}(x)}{x}\right| \leq 10^{-n+1}.$$

In a computing device that uses n-digit chopping (decimal) arithmetic, the floating-point approximation of a positive real number x is denoted by f(x). The approximation is done by chopping off the digits after the nth decimal place.

To prove that
| x- f(x) | / |x| ≤ 10^(-n+1)

We know that the chopping process removes the digits after the nth decimal place. So, the maximum absolute error of the approximation is the greatest possible value of the removed digits.

The greatest possible value of the removed digits is 0.5 x 10^(-n) because the removed digits can be any value between 0 and 9 and the greatest possible value is 9 x 10^(-n), which is rounded to 0.5 x 10^(-n) for the upper bound.

Now, the relative error of the approximation is the absolute error divided by the original value.

| x- f(x) | / |x| = (0.5 x 10^(-n)) / |x|

As x is positive, |x| = x and the relative error is less than or equal to 0.5 x 10^(-n)

And as the relative error is less than or equal to 0.5 x 10^(-n) and 10^(-n) = 10^(-n+1) so the relative error is less than or equal to 10^(-n+1)

Therefore, if the device uses n-digit chopping (decimal) arithmetic, then | x- f(x) | / |x| ≤ 10−n+1.

6.

The ideal gas law is given by $PV = nRT$ where R is the gas constant. We are interested in knowing the value of T for which $P = V = n = 1$. If R is known only approximately as $R_A = 8.3143$ with an absolute error at most 0.12×10^{-2}. Obtain an upper bound for the absolute relative error in the computation of T that results in using R_A instead of R?

The ideal gas law is given by P V = nRT where R is the gas constant. We are interested in finding the value of T for which P = V = n = 1. If R is known only approximately as RA = 8.3143 with an absolute error at most 0.12 × 10^(-2).

To obtain an upper bound for the absolute relative error in the computation of T, we can use the formula for relative error:

|(T-TR)|/T <= |(R-RA)/RA|

Where T is the true value of T, TR is the approximation of T using RA and R is the true value of the gas constant.

Given that the absolute error in RA is at most 0.12 × 10^(-2), we have
|R-RA| <= 0.12 × 10^(-2)

So,
|(R-RA)/RA| <= 0.12 × 10^(-2) / 8.3143 = 0.014

Therefore, the upper bound for the absolute relative error in the computation of T that results in using RA instead of R is 0.014.

7.

Let $x_A = 3.14$ and $y_A = 2.651$ be obtained from x_T and y_T using 4-digit rounding. Find the smallest interval that contains

(i) x_T (ii) y_T (iii) $x_T + y_T$ (iv) $x_T - y_T$ (v) $x_T \times y_T$ (vi) x_T/y_T.

i)

The smallest interval that contains xT would be [xA - δ, xA + δ], where δ is the machine epsilon. In this case, the machine epsilon for 4-digit rounding is 10^(-4) = 0.0001, so the smallest interval that contains xT is [3.14 - 0.0001, 3.14 + 0.0001] = [3.1399, 3.1401].

Ii) similarly for YT = [2.6509,2.6511]

Other you can check by yourself

8.

Obtain the number of significant digits of $x_A = 0.025678$ present in $x = 0.025611$.

soln. | x - xA | / |x | = 0.0026 = 0.2 * 10^-2 <= 0.5*10^-2
r = 2+1 = 3

9.

Instead of using the true values $x_T = 0.62457371$ and $y_T = 0.62457238$ in calculating $z_T = x_T - y_T$, if we use the approximate values $x_A = 0.62451251$ and $y_A = 0.62458125$, and calculate $z_A = x_A - y_A$, then find the loss of significant digits in the process of calculating z_A when compared to the significant digits in x_A.

Similar as above, first find z, zA then use above formula to find loss of significant digit

10.

Let $x_A = 0.04078$ has exactly 3 significant digits with respect to the real number x_T. Find the smallest interval in which x_T lies.

| x - 0.04078 |/ |X| <= 0.5 * 10^-2
| 1 - 0.04078/x | <= 0.5 * 10^-2
-0.005 <= 1 - 0.04078/x <= 0.005
-1.005<< - 0.04078/x <= -0.995
1.005>= 0.04078/x and 0.04078/x >= 0.995
X>=0.0405771 and x <= 0.0409845

11.

Given $x = 0.75371$ and $y = -0.49572$. Let the product $x * y$ be computed using 3-digit rounding floating-point arithmetic. What is the absolute value of the total error? [Give the final answer with at least 6-digits after decimal places]

Given x = 0.75371 and y = -0.49572, let the product x*y be computed using 3-digit rounding floating-point arithmetic.

To compute the product using 3-digit rounding, we round x and y to the third decimal place. xA = 0.754 and yA = -0.496.

The product xA*yA = 0.754 * -0.496 = -0.37344

The true value of the product x*y = 0.75371 * -0.49572 = -0.374066292

The absolute value of the total error is |-0.374066292 - (-0.37344)| = 0.000066292

12.

For small values of x, the approximation $\sin x \approx x$ is often used. Obtain a range of values of x for which the approximation gives an absolute error of at most $\frac{1}{2} \times 10^{-6}$.

The approximation sin x ≈ x is often used for small values of x. To obtain a range of values of x for which this approximation gives an absolute error of at most 1/2*10^-6, we can use the following:

|sin x - x| <= 1/2*10^-6

We know that |sin x| <= 1, so we can simplify the inequality as:

|x - sin x| <= 1/2*10^-6

The range of values of x for which the approximation gives an absolute error of at most 1/2*10^-6 is the range of values of x for which the above inequality is true.

We can find the range of values of x for which the above inequality is true by evaluating it for different values of x.

For x = 0, the inequality is true.

For x = 0.000001, we have |x - sin x| = |0.000001 - 0.000001| = 0 which is less than or equal to 1/2*10^-6, so the inequality is true.

For x = 0.00001, we have |x - sin x| = |0.00001 - 0.00001| = 0 which is less than or equal to 1/2*10^-6, so the inequality is true.

For x = 0.0001, we have |x - sin x| = |0.0001 - 0.0001| = 0 which is less than or equal to 1/2*10^-6, so the inequality is true.

And when we try larger value of x, we can see that the inequality is no longer true.

So the range of values of x for which the approximation gives an absolute error of at most 1/2*10^-6 is x in the range [0, 0.0001].

So the range of values of x for which the approximation sin x ≈ x gives an absolute error of at most 1/2*10^-6 is [0, 0.0001]

This solution may or may not be right

13.

Is the process of computing the value of the function $f(x) = (e^x - 1)/x$ stable or unstable for $x \approx 0$? Justify your answer.

Check stepwise step condition which is given in reading material part 2