1. What is a truncation error in floating-point notation, and how does it occur?

Answer

A truncation error in floating-point notation refers to the loss of precision or accuracy when representing a real number in a limited number of binary digits. It occurs when the mantissa field, which stores the fractional part of the number, doesn't have enough bits to accommodate the complete value. For example, if we try to store 25⁄8 in a one-byte floating-point system, we encounter a truncation error. When converting 25⁄8 to binary (10.101), the rightmost 1 (representing 1⁄8) cannot fit in the available space, resulting in a loss of information. This error arises because the mantissa field is insufficiently large to hold the entire value, causing the truncated bits to be discarded.

2. How can truncation errors in floating-point notation be mitigated or reduced?

Answer

: Truncation errors in floating-point notation can be mitigated by increasing the size of the mantissa field. Most modern computers use longer mantissa fields, such as 32 bits, to store values in floating-point notation, allowing for greater precision. A longer exponent field is often used in conjunction with a longer mantissa field. This approach reduces truncation errors by providing more bits for representing the fractional part of a number accurately.

3. What is the difference between nonterminating expansions in binary and decimal notation, and how does this relate to truncation errors?

Answer (150 words): Nonterminating expansions refer to numbers that cannot be expressed with a finite number of digits in a given base. In binary notation, more values have nonterminating representations compared to decimal notation. For instance, the decimal value 1/10 is nonterminating in binary. This difference is significant when dealing with truncation

errors in floating-point notation. For example, when storing monetary values like dollars and cents using floating-point notation, the value of a dime (1/10 of a dollar) cannot be accurately represented because 1/10 is nonterminating in binary. To address this, data can be manipulated in units like pennies, ensuring that all values are integers, which can be stored accurately using two's complement or similar methods.

4. Explain how the order of adding small quantities affects the accuracy of their representation in floating-point notation.

Answer

The order of adding small quantities in floating-point notation can significantly impact the accuracy of their representation. When adding small quantities, it's advisable to start with the smallest and progressively add larger ones. This approach minimizes the potential for truncation errors. For example, consider adding 1/8 to 1/8 and then adding the result to 2½. If we add 1/8 to 1/8 first, we get ¼, which can be stored accurately. Adding this to 2½ results in an accurate representation of 2¾. However, if we added 2½ and 1/8 first, we'd lose the precision of the 1/8 due to truncation errors.

These questions and answers provide a comprehensive understanding of truncation errors in floating-point notation as described in the provided paragraph.