Minimizing Fractional Errors in Floating Point Number

Abstract:

Programming languages support floating point numbers as a built-in data type. However the representation of floating point numbers differs from one language to another. Integer division may result in floating point value that in many cases can be fully accommodated in computer memory or it goes infinitely which produces a lost in the output value. If the result is then taken for further calculations the gape of the error increases and the effect may not be acceptable. In this research a solution is proposed based on representing fractional numbers (mantissa, base and exponent) using integer numbers format. In this case we get three parts: the Integer value, the Numerator value and the Denominator value. The operations on such representation are also introduced. This representation is evaluated and compared with the traditional (Mantissa) representation. Sample programs using C++ language are developed showing how this representation can be used and how fractional numbers are declared and used, with comparison with the normal representation.