What are some other software systems that should be careful about (or entirely avoid) using floating points? How might their use of floating points create a situation that's harmful to the intended users of the system?

Aviation Software

Aviation software, such as autoland or autopilot systems, often depend heavily on highly precise calculations of altitude, speed, and trajectory. Hence, the accumulation of small rounding errors in these programs can have catastrophic consequences.

For example, in an autoland system, rounding errors may cause the program to misinterpret the aircraft's altitude, thinking it is higher or lower than it actually is. A miscalculation by even a relatively small margin, say ten metres, could result in a crash landing.

Small rounding errors could cause similarly disastrous consequences in missile guiding systems or air traffic control systems.

Medical Software

Medical software, which often involves continuous monitoring of vital signs and the precise administration of treatments, must be particularly cautious with floating point arithmetic.

For example, in systems like pacemakers, for instance, even small timing errors due to rounding inaccuracies can be fatal. Storing time as an integer would likely help to combat this inaccuracy. To achieve this a very small unit of time, such as picoseconds or even smaller, should be used.

Similarly, software that calculates drug dosages, radiation therapy, or insulin levels in diabetic patients requires extremely precise computation. For example, a small rounding error in the calculation of radiation dosage could result in too much exposure, leading to tissue damage, or too little, making the treatment ineffective.