CPS 101 Introduction to Computational Science

Wensheng Shen

Department of Computational Science
SUNY Brockport
Chapter 6 Modeling Process

- Definition
- Model classification
- Steps of modeling
6.1 Definition

- Modeling is the application of methods to solve complex, real-world problems in order to make predictions about what might happen with different conditions.
Examples

- Dr. Jerrold Marsden, a computational physicist at CalTech, models space mission trajectory design;
- Dr. Julianne Collins, a genetic epidemiologist at the Greenwood Genetics Center, models epidemiological studies using SAS;
- People at the Human-Technology Interaction Center of The University of Oklahoma models strategic planning in air traffic control;
- Scientists use computational models to discover land mines;
- Scientists model the oscillating patterns of the disease dengue fever.
6.2 Modeling classification

- **Stochastic versus deterministic**
  - A stochastic (probabilistic) model exhibits random effects --- the path of a hurricane;
  - A deterministic model does not exhibit random effects --- a falling object in a vacuum.

- **Static versus dynamic**
  - A static model is not involved in time --- the volume of a tank;
  - A dynamic model changes with time --- the number of fish in a pond;

- **Continuous versus discrete**
  - A continuous model changes continuously with time and space;
  - A discrete model changes with time and space in a discrete sequence;
6.3 Steps of modeling

- Analyze the problem
- Formulate a model
  - Gather data
  - Determine variables and units
  - Establish relationships among variables and submodels
  - Determine equations and functions
- Solve the model
- Verify and interpret the model’s solution
  - Verification: to make sure solving the problem correctly
  - Validation: to make sure solving the right problem
- Report the model
Chapter 7 Errors

- Errors occur at any stage in computational science, from data collection to computer implementation. It is very important that a modeler must know where errors are from and how to prevent them.
Error sources

- Data errors
  - A sensor might malfunction
  - The accuracy of sensor is not sufficient
  - The sensor is not calibrated or misread

- Modeling errors
  - Error in formulating models. Lord Kevin developed a model to calculate the Earth age to be 20 to 40 million years old based on the assumption the Earth was cooling from a molten mass with the Sun being the only source of energy.

- Implementation errors
  - Computational scientist could make errors. For example, in 1999 NASA’s Mars Climate Orbiter spacecraft was lost because the builder of the spacecraft, Lockheed Martin Corp., programmed it to use English units, but NASA’s Jet Propulsion scientists employed metric units.
7.1 Precision

- Precision: the precision of a number is the number of significant digits.
- Significant digits of an integer are all the digits except the leading and trailing zeros. The significant digits of a float point are all the digits except the leading zeros.
- The precision of 0.0055600 is 5, the precision of 005,664,080 is 6.
- A normalized number in exponential notation has the decimal point immediately preceding the first nonzero digit, such as $0.98436 \times 10^3$. The significand is 98436 and the exponent is 3.
7.2 Absolute and relative errors

- **Absolute error**: the absolute value of the difference between the exact solution and the computed results.

  \[
  \text{absolute error} = |\text{exact} - \text{computed}|
  \]

- **Relative error**: the ratio between the absolute error and the absolute value of the exact solution.
Example: the exact solution of $2 \times 2$ is 4, but the computer gives a result of 3.9999999999.

- The absolute error = 0.0000000001.
- The relative error = 0.000000000025.
7.3 Round-off error

- Round-off error is the error caused by rounding a number to a certain precision. Round 0.3364 to 0.336.

- To round a normalized number to precision $k$, consider the $(k+1)$th significant digit. If $d$ is less than 5, round down by truncating the significand to $k$ significant digits. If $d$ is greater than or equal to 5, round up by truncating the significand to $k$ significant digits and then adding 1 to the $k$th significant digit of the significand.

- Round-off error occurs in computers when there are not enough bits to store an entire floating point number and it is approximated to the nearest number that can be represented. $\frac{1}{3} = 0.33333333333...$
Assignment statement: \( x = 1.0 / 3.0 \), gives the value of the expression on the right to the variable \( x \) on the left.

An assignment statement causes the computer to store the value of an expression in a memory location associated with a variable.
Overflow is an error condition that occurs when there are not enough bits to express a value in a computer. Overflow happens when we add to positive numbers to get a negative result, or when we add two negative numbers to get a positive result.

Suppose a very small computer uses 16 bits to store an integer. If we ask the computer to perform the sum 20480 + 16384, the result will be surprisingly a negative number, -28672. This is because when the leftmost bit, the sign bit, gets a carry from the addition on the right, converting the result to a negative number.
Result: -28672, Why?

Two’s complement: \(-2^{15} + 2^{12} = -28672\)

\[2^{15} = 32768\]
In 1996, an overflow error caused the European Space Agency’s Ariane 5 rocket to explode. In less than 37 seconds after the launch, the guidance system’s computer attempted to convert the rocket’s sideways velocity from a 64-bit floating point number to a 16-bit integer. Because the number was too large, overflow results, and the guidance system attempted a severe correction for a wrong turn that had not occurred. Very quickly the rocket had to self-destruct.

The overflow of a few bits caused the loss of a rocket that took 10 years and $7 billion to develop.

http://www.around.com/ariane.html
Underflow is an error condition that occurs when the result of a computation is too small for a computer to represent.

Example: suppose a computer can express the smallest floating point number with the magnitude of $10^{-39}$. If in an arithmetic operation, we obtained a correct value of $10^{-48}$, which is smaller than the smallest floating-point value the computer can represent, then underflow occurs, and the computer evaluates $10^{-48}$ to zero.
7.5 Arithmetic errors

- Errors can arise in addition. Consider \((0.684 \times 10^3) + (0.950 \times 10^{-2})\), we have 684.0095. If our computer allows for only three significant digits, the normalized result is \(0.684 \times 10^3\), and the effect of the \(0.950 \times 10^{-2}\) is lost.

- Strategy: when adding numbers whose magnitude are dramatically different, we should accumulate smaller numbers first before combining them with larger ones.

- Similarly, when multiplying and dividing in a term, to avoid loss of precision, we usually should perform all multiplications in the numerator before dividing by the denominator.
7.6 Error propagation

- Error can propagate in loops, code segments that are executed repeatedly. Round-off errors of floating point operations accumulate within loops.

- Example: During the Gulf War in Dharan, Saudi Arabia, an American Patriot Missile battery failed to intercept a Scud missile due to accumulated errors. The Scud hit an American Army barracks, killed 28 soldiers, and injured more than 100 others. The Patriot’s internal computer clock measured time in tenths of a second, and the actual time can be obtained by multiplying the number of ticks by 1/10. For example, 15 ticks indicate an elapsed time of (15 ticks)(0.1 sec/tick). The computer in the Patriot missile used 24 bits to store numbers. However, because 1/10 has an infinite expansion in binary representation, the system could not hold all the number’s bits. Each one-tenth increment produced an error of about 0.000000095 sec. At the time of the disaster, the Patriot Missile had been operating for 100 hours, causing an error of about (100 hrs)(60 min/hr)(60 sec/min)(10 ticks/sec)(0.00000095 sec/tick)=0.34 sec. A Scud flew with an velocity of 1,676 meter/sec, so it could fly more than a half kilometer in 0.34 sec. The intercepting Patriot Missile missed its target.

The number $1/10$ equals
$1/2^4 + 1/2^5 + 1/2^8 + 1/2^9 + 1/2^{12} + 1/2^{13} + \ldots$. In other words, the binary expansion of $1/10$ is
$0.00011001100110011001100110011001100\ldots$. Now the 24 bit register in the Patriot stored instead
$0.00011001100110011001100110011001100\ldots$ introducing an error of
$0.0000000000000000000000000000000000\ldots$ binary, or about $0.000000095$ decimal. Multiplying by the number of tenths of a second in 100 hours gives
$0.000000095 \times 100 \times 60 \times 60 \times 10 = 0.34$.

During that third of a second, a Scud flew about 1,676 meters, so that the intercepting Patriot Missile missed its target.
7.7 Other errors

- Comparison of floating point numbers:
  - if(x==z) // do not test floating point numbers this way, because of the conversion from decimal number to binary numbers.
- Truncation error: a truncation error occurs when a truncated, or finite, sum is used as an approximation for the sum of an infinite series.
- Example:

\[
e^x = 1 + x + \frac{x^2}{1 \cdot 2} + \frac{x^3}{1 \cdot 2 \cdot 3} + \ldots + \frac{x^n}{n!} + \ldots
\]