

Computational Methods

CMSC/AMSC/MAPL 460

Representing numbers in floating point

Ramani Duraiswami,

Dept. of Computer Science

Effects of floating point errors

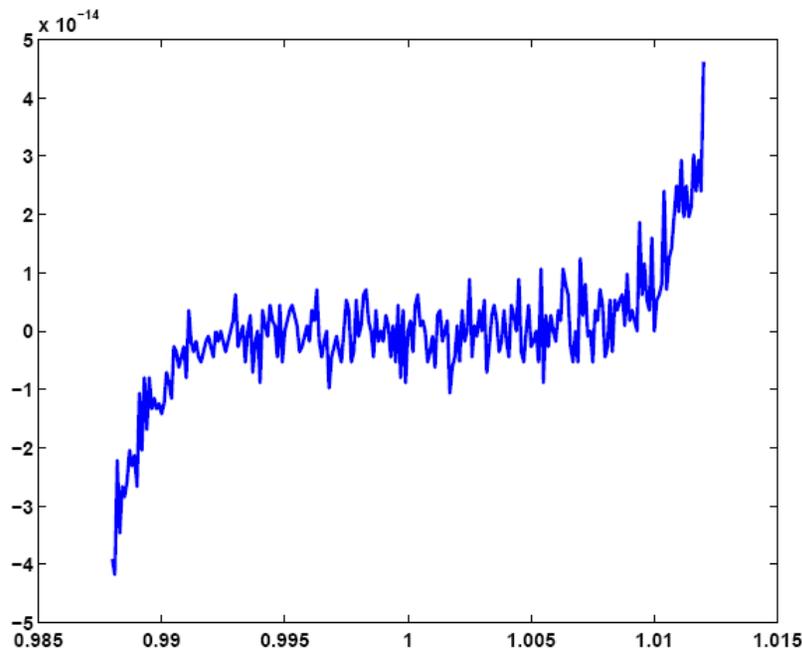
- Singular equations will only be nearly singular
- Severe cancellation errors can occur

$$\begin{aligned}17x_1 + 5x_2 &= 22 \\ 1.7x_1 + 0.5x_2 &= 2.2\end{aligned}$$

$$\begin{aligned}A &= [17 \ 5; \ 1.7 \ 0.5] \\ b &= [22; \ 2.2] \\ x &= A \setminus b\end{aligned}$$

```
x = 0.988:.0001:1.012;
```

```
y = x.^7-7*x.^6+21*x.^5-35*x.^4+35*x.^3-21*x.^2+7*x-1; produce  
plot(x,y)
```



$$\begin{aligned}x &= \\ &-1.0588 \\ &8.0000\end{aligned}$$

Class Outline

- Computations should be as accurate and as error-free as possible
- Sources of error:
 - Poor models of a physical situation
 - Ill-posed problems
 - Errors due to representation of numbers on a computer and successive operations with these
- How do we analyze an algorithm for correctness?
 - Forward error analysis
 - Backward error analysis
- Well posedness

Error

- What we need to know about error:
 - how does error arise
 - how machines do arithmetic
 - fixed point arithmetic
 - floating point arithmetic
 - how errors are propagated in calculations.
 - how to measure error

Typical task that uses scientific computing

- Evaluate safety of a machine part
- Tasks
 1. Measure the parts dimensions, shape etc. and discretize it (e.g., via finite elements)
 2. Determine the material it is made of
 3. Find the mathematical models (equations) that determine how the part will deform according to loads
 4. Discretize the equations (e.g., via finite elements)
 5. Solve it on the computer

Errors

- Each step is characterized by some error
 1. Measurement errors:
 2. Errors in properties
 3. Inexact mathematical models
 4. Discretization errors: something continuous is represented discretely
 5. Errors in the solution to discrete representations of numbers

Errors are inevitable

- Everybody did the best they could
- No one made any mistakes, yet answer could be wrong
- Goal of error analysis is to
 - determine when the answer can be relied upon
 - Which algorithms can be trusted for which data
- Last class we focused on errors due to the finite representation of numbers

Modeling

- Original mathematical models may be poorly specified or unavailable
 - E.g. Newton's laws work for non relativistic dynamics
 - Turbulence
 - ...
- Computing with a poor model will lead to inevitable errors
- Quantities that are measured may be done so with error and bias
 - Using them in computation will lead to errors
- Approaches to fix these errors are in the domain of statistics
 - Will not be much discussed in this course

Numerical Modeling and Measurement Errors

- Continuous mathematical models have to be represented in discrete form on the computer
 - Finite-difference or finite-element discretization
 - Continuous quantities may be represented using linear interpolants
 - Model may only reach accurate answer in the limit
 - Round-off errors – continuous numbers represented with discrete representations on the computer

Error definition

- Computation should have result: c
- Actual result is: x
- $\text{abs_err} = |x - c|$
- $\text{rel_err} = |x - c| / x$ for $x \neq 0$
- $x = (c) \times (1 + \text{rel_err})$

- Usually we do not know c
 - No need to solve the problem if we already knew it!
- Error analysis tries to estimate abs_err and rel_err using computed result x and knowledge of the algorithm and data

Two issues

- Can we design algorithms to minimize errors
- Can we estimate errors based on knowledge of algorithm
 - Error Analysis – forward and backward

Errors due to round off in addition

- Errors can be magnified during computation.
- Example: 2.003×10^0 (suppose $\pm .001$ or .05% error)
- 2.000×10^0 (suppose $\pm .001$ or .05% error)
- Result of subtraction: 0.003×10^0
- but true answer could be as small as $2.002 - 2.001 = 0.001$, or as large as $2.004 - 1.999 = 0.005!$
- So error in the answer is as much as ($\pm .002$ or 200% error if true answer is 0.001)
- Called: **Catastrophic cancellation**, or “loss of significance”

Addition:

- We could generalize this example to prove a theorem:
- When **adding or subtracting**, the bounds on **absolute errors** add.

Multiplication/Division

- What if we multiply or divide?
- Suppose x and y are the true values, and X and Y are our approximations to them. If

$$X = x (1 - r) \text{ and } Y = y (1 - s)$$

then r is the relative error in x and s is the relative error in y .

Can show that
$$\left| \frac{xy - XY}{xy} \right| \leq |r| + |s| + |rs|$$

- If r and s are small, then we can ignore $|rs|$ term

Rules of thumb

- Addition/subtraction: Bounds on **absolute errors** add
- Multiplication/Division: Bounds on **relative errors** add
- One way to analyze the algorithm is to assume, this error occurs in each arithmetic operation
- Worst case analysis
- Such error bounds (forward error bounds) are often pessimistic

Error Analysis

- Two primary techniques of error analysis
 - Forward Error Analysis
 - Floating-point representation of the error is subjected to the same mathematical operations as the data itself.
 - Equation for the error itself
 - Backward Error Analysis
 - Attempt to regenerate the original mathematical problem from previously computed solutions
 - Minimizes error generation and propagation

Error Analysis

- Forward and Backward error analysis
- Forward error analysis
 - Assume that the problem we are solving is exactly specified
 - Produce an approximate answer using the algorithm considered

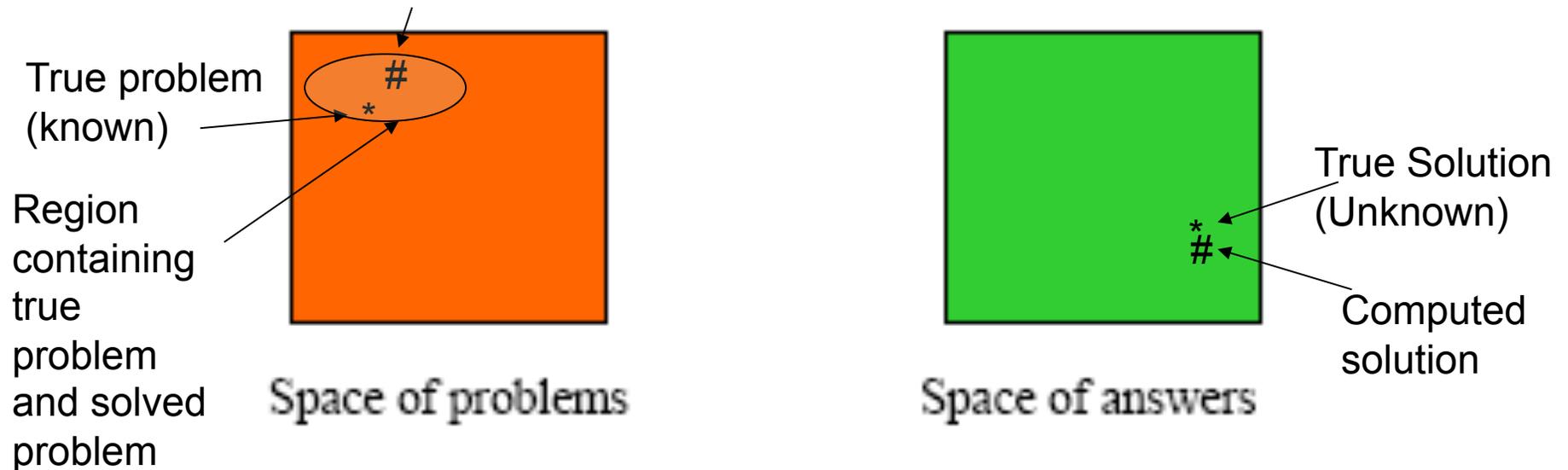


- Goal of forward error analysis produce region guaranteed to contain true soln.
- Report region and computed solution

Backward error analysis

- We know that our problem specification itself has error (“error in initial data”)
- So while we think we are solving one problem we are actually solving another

Unknown problem
(the one actually solved)



- Given an answer, determine how close the problem actually solved is to the given problem.
- Report solution and input region

Testing for Error Propagation

- Use the computed solution in the original problem and check if it satisfies it
- Use Double or Extended Precision rather than Single Precision
- Rerun the problem with slightly modified (incorrect) data and look at the results

Well posed problems

- Hadamard postulated that for a problem to be “well posed”
 1. *Solution must exist*
 2. *It must be unique*
 3. *Small changes to input data should cause small changes to solution*
- Essentially this means the regions in the problem space and solution space must be small