

Homework 1

CS 210

Andres Calderon
SID 861243796

April 12, 2016

Question	Points	Score
1	10	
2	5	
3	5	
4	5	
5	10	
6	10	
7	10	
8	15	
9	15	
10	15	
Total	100	

Errors and sources of error

- (Heath 1.2) What are the approximate absolute and relative errors in approximating π by each of the following quantities?
 - 3
 - 3.14
 - $22/7$

Answer: we have $\pi(3.141592)$, to 6 decimal places, then:

- Absolute error: $3 - \pi = -0.141592$
Relative error: $\frac{-0.141592}{\pi} = 0.045070$
 - Absolute error: $3.14 - \pi = -0.001592$
Relative error: $\frac{-0.001592}{\pi} = 0.00050$
 - Absolute error: $\frac{22}{7} - \pi = 0.001265$
Relative error: $\frac{0.001265}{\pi} = -0.00040$
- (Heath 1.3) If a is an approximate value for a quantity whose true value is t , and a has a relative error r , prove from the definitions of these terms that $a = t(1 + r)$.

Answer:

$$\begin{aligned} absolute_error &= a - t \\ r &= \frac{absolute_error}{t} \\ r &= \frac{a-t}{t} \\ tr &= a - t \\ t + tr &= a \\ t(1+r) &= a \end{aligned}$$

3. For each of the following statements, indicate whether the statement is true or false.

T/**(F)** The use of a floating point number system results in so-called truncation error in numerical methods.

(T)/**F** It is possible for small rounding errors to catastrophically destroy the accuracy of an algorithm.

T/**(F)** A large absolute error implies a large relative error.

Conditioning and stability

4. For each of the following statements, indicate whether the statement is true or false.

(T)/**F** A problem is ill-conditioned if its solution is highly sensitive to changes in its data.

T/**(F)** We can improve conditioning of a problem by switching from single to double precision arithmetic.

(T)/**F** In order to numerically solve a problem accurately, it is necessary to have both a well-conditioned problem and a stable algorithm.

(T)/**F** A condition number of 1 means the problem is well-conditioned.

5. (Heath 1.5) Consider the function $f : \mathbb{R} \rightarrow \mathbb{R}$ defined by $f(x, y) = x - y$. Measuring the size of the input (x, y) by $|x| + |y|$, and assuming that $|x| + |y| \approx 1$ and $x - y \approx \epsilon$, show that $\text{cond}(f) \approx 1/\epsilon$. What can you conclude about the sensitivity of subtraction?

Answer:

Given $f(x, y) = x - y$ its absolute error is:

$$absolute_error = f(x + \Delta x, y + \Delta y) - f(x, y) \approx \Delta x f'_x(x, y) + \Delta y f'_y(x, y)$$

Then, the relative error is:

$$relative_error = \frac{f(x + \Delta x, y + \Delta y) - f(x, y)}{f(x, y)} \approx \frac{\Delta x f'_x(x, y) + \Delta y f'_y(x, y)}{f(x, y)}$$

As $f(x, y) = x - y \approx \epsilon$, then:

$$relative_error \approx \frac{\Delta x f'_x(x, y) + \Delta y f'_y(x, y)}{\epsilon}$$

Finally, the condition number is:

$$Cond \approx \left| \frac{\frac{\Delta x f'_x(x, y) + \Delta y f'_y(x, y)}{\epsilon}}{\frac{\Delta x + \Delta y}{|x| + |y|}} \right|$$

As $|x| + |y| \approx 1$, then:

$$Cond \approx \left| \frac{\Delta x f'_x(x, y) + \Delta y f'_y(x, y)}{\epsilon(\Delta x + \Delta y)} \right|$$

Since $f'_x(x, y) = 1$ and $f'_y(x, y) = -1$, we have:

$$Cond \approx \left| \frac{\Delta x - \Delta y}{\epsilon(\Delta x + \Delta y)} \right| \approx \frac{1}{\epsilon}$$

As $Cond \approx \frac{1}{\epsilon}$, we can conclude that the subtraction is very sensitive to small changes in the input.

6. Show that for a differentiable function $f(x)$, the sensitivity of evaluating the function is described by the condition number

$$\kappa_f \approx \left| \frac{xf'(x)}{f(x)} \right|,$$

and that the sensitivity of evaluating the inverse function $g(y)$ is given by

$$\kappa_g = \frac{1}{\kappa_f}.$$

Answer:

To find κ_f from $f(x)$ we have to compute the absolute error:

$$absolute_error = f(x + \Delta x) - f(x) \approx \Delta x f'(x)$$

Then, the relative error:

$$relative_error = \frac{f(x + \Delta x) - f(x)}{f(x)} \approx \frac{\Delta x f'(x)}{f(x)}$$

and finally, the condition number:

$$\kappa_f \approx \left| \frac{\frac{\Delta x f'(x)}{f(x)}}{\frac{\Delta x}{x}} \right| \approx \left| \frac{\Delta x f'(x)x}{f(x)\Delta x} \right| \approx \left| \frac{xf'(x)}{f(x)} \right|$$

For the second part, I assume $g(y) = \frac{1}{f(x)}$ so, $g'(y) = \frac{1}{f'(x)}$ and $y = \frac{1}{x}$, then:

$$\kappa_g \approx \left| y \frac{g'(y)}{g(y)} \right| \approx \left| \frac{1}{x} \frac{\frac{1}{f'(x)}}{\frac{1}{f(x)}} \right| \approx \left| \frac{f(x)}{xf'(x)} \right| \approx \frac{1}{\kappa_f}$$

Floating point

7. (Heath 1.7) A floating point number system is characterized by four integers: the base β , the precision p , and the lower and upper limits L and U of the exponent range.
 - (a) If $\beta = 10$, what are the smallest values of p and U , and the largest value of L , such that both 2365.27 and 0.0000512 can be represented *exactly* in a *normalized* floating-point system?
 - (b) How would your answer change if the system is not normalized, i.e., if gradual underflow is allowed?
8. (Trefethen & Bau 13.1) Between an adjacent pair of nonzero IEEE single precision real numbers, how many IEEE double precision numbers are there?
9. For each of the following statements, indicate whether the statement is true or false.

T/F If two numbers are exactly representable in floating point, then the result of an arithmetic operation on them is also an exactly representable floating point number.

T/F Floating point arithmetic is commutative, but not associative.

T/F Floating point numbers are distributed uniformly throughout their range.

T/F In a denormalized floating point system, the representation of a number is unique.

T/F Addition of two positive floating point numbers may cause underflow.

T/F Division of two positive floating point numbers may cause overflow.

T/F Denormalization is used to mitigate arithmetic overflow.

T/F In a normalized floating point system, the representation of a machine number is not unique.

10. *Computer problem* (Heath 1.10)

- (a) Write a program to solve the quadratic equation $ax^2 + bx + c = 0$ using the standard quadratic formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

or the alternative formula

$$x = \frac{2c}{-b \mp \sqrt{b^2 - 4ac}}.$$

Your program should accept values for the coefficients a , b , and c as input and produce the two roots of the equation as output. Your program should detect when the roots are not real, but need not use complex arithmetic explicitly (for example, you could return the real and imaginary parts of the complex conjugate roots in this case). You should guard against unnecessary overflow, underflow, and cancellation. Try to make your program robust when given unusual input values, such as $a = 0$ or $c = 0$, which otherwise would make one of the formulas fail. Any root that is within the range of the floating-point system should be computed accurately, even if the other is out of range. Submit a copy of your code. You may use the language of your choice.

- (b) When should you use each of the two formulas?
 (c) Test your program using the following values for the coefficients and give your results:

a	b	c
6	5	-4
6×10^{154}	5×10^{154}	-4×10^{154}
0	1	1
1	-10^5	1
1	-4	3.999999
10^{-155}	-10^{155}	10^{155}