Errors in Numerical Analysis Calculations – Prof. Richard B. Goldstein

SOURCES OF ERRORS

(1) Measurement Errors – laboratory / machine reading errors – one cannot expect the numerical analysis results to be more accurate than the data it came from

(2) Truncation Errors – calculus operations / series that are terminated after a few terms - there is a limit to how well any polynomial can approximate a function that is represented by an infinite power series

(3) Roundoff and Chopping Errors – numerical computations that involve limited storage space of numbers – computers typically store numbers as 4, 5, or 6 bytes when numbers such as pi or the square root of 2 and even simple fractions such as 1/7 have infinite representations in decimal, binary, or hexadecimal

TEXT EXAMPLE

[1] *Nesting* – reduces the number of operations and improves accuracy

$f(x) = x^3 - 6.1x^2 + 3.2x + 1.5$ at x = 4.71

Exact f(4.71) = 104.487111 – 6.1(22.1841) + 3.2(4.71) + 1.5 = -14.263899

3-D Chop f(4.71) = 104 – 6.1(22.1) +3.2(4.71) + 1.5
= 104 – 134 +15.0 + 1.5
= -13.5

3-D Round f(4.71) = 105 – 6.1(22.2) + 3.2(4.71) + 1.5 = -13.4 note x^3 = x x x is used

Nesting ((x – 6.1)x + 3.2)x + 1.5

3-D Chop f(4.71) = ((4.71 – 6.1)4.71 + 3.2)4.71 + 1.5 = -14.2

3-D Round f(4.71) = … = -14.3 (best)

- Three decimal chopping or rounding requires the result after each operation to be chopped or rounded to fit into three places. Both answers were reduced to less than three digits after several operations. When three digit chopping and rounding were used the results were accurate within the third digit.

MORE ERRORS IN ARITHMETIC PROCESSES

		with chopping	with rounding
[2]	$x = 0.6532849 \times 10^2$	$x^* = 0.6532 \times 10^2$	$x^{**} = 0.6533 \times 10^2$
	$y = 0.6531212 \times 10^2$	$y^* = 0.6531 \times 10^2$	$y^{**} = 0.6531 \times 10^2$

$x - y = 0.0001637 \times 10^2 \rightarrow 0.1637 \times 10^{-1}$

$x^* - y^* = 0.0001 \times 10^2 = 0.1000 \times 10^{-1}$

$x^{**} - y^{**} = 0.0002 \times 10^2 = 0.2000 \times 10^{-1}$

- although rounding did better than chopping, there was still a great deal of loss of accuracy in the subtraction process of two close numbers

[3] $0.9621 \times 10^0 + 0.6732 \times 10^0 = 1.6353 \times 10^0 \rightarrow 0.1635 \times 10^1$

- not much of a loss when two almost equal numbers are added

[4] $0.9621 \times 10^M + 0.6732 \times 10^M \rightarrow$ overflow

[5] $0.5055 \times 10^4 + \underbrace{0.4000 \times 10^0 + \ldots + 0.4000 \times 10^0}_{\text{added 11 times}} \rightarrow 0.5055 \times 10^4$

because $0.5055 \times 10^4 + 0.4000 \times 10^0 = 0.50554 \times 10^4 \rightarrow 0.5055 \times 10^4$ each time reacts as if one is adding zero

however, $0.4000 \times 10^0 + \ldots + 0.4000 \times 10^0 + 0.5055 \times 10^4 = 4.4 \times 10^0 + 0.5055 \times 10^4$ or $0.50594 \times 10^4 \rightarrow 0.5059 \times 10^4$

- it is better to add small terms first and then add the larger number
- addition is *not* always associative on a computer

[6] $$\sum_{i=1}^{999} \frac{1}{k(k+1)} = 0.999 \text{ exactly}$$

		absolute error
forwards	0.998 970 9	0.000 028 1
backwards	0.998 999 2	0.000 000 8

- similar to [5] – it is better to add a series from the smallest to largest terms - the forward addition of a thousand 7 decimal digit accuracy numbers in this series produced an answer accurate to 4 decimal digits but the backward addition (smaller numbers first) produced an answer accurate to 6 decimal digits

[7] $x^2 + 62.10x + 1 = 0$ has roots -62.08389… and -0.01611…

$$x = \frac{-62.10 + \sqrt{(62.10)^2 - 4(1)(1)}}{2} = \frac{-62.10 + \sqrt{(3856.41 - 4}}{2} \Rightarrow \frac{-62.10 + \sqrt{3852}}{2}$$

$$x = \frac{-62.10 + 62.06}{2} = \frac{-0.04}{2} = -0.0200$$ **note**: subtraction of two close numbers

An alternative quadratic formula to $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ is $\frac{-2c}{b \pm \sqrt{b^2 - 4ac}}$ which gives $\frac{-2}{62.10 \pm 62.06}$ yielding $\frac{-2}{124.2} = -0.0161$ which is excellent but also $\frac{-2}{0.04} = -50$ which is a poor second root

- rewriting the quadratic equation helped with one root but hurt with the other root; the problem again is with the subtraction of two close numbers wiping out the accuracy

GENERAL CONCLUSIONS

- **Rounding usually does better than chopping.**
- **Nesting improves polynomial evaluation.**
- **Add terms from smallest to largest in a series.**
- **Avoid subtracting numbers that are close by rewriting an equation algebraically or by rearranging the terms.**