Directions: Show your work! Answers without justification will likely result in few points. Your written work also allows me the option of giving you partial credit in the event of an incorrect final answer (but good reasoning). Indicate clearly your answer to each problem (e.g., put a box around it). You should skip 20 points (of your choice). Write "skip" **clearly** on those parts you skip. **Good luck!**

Problem 1. (20 pts) Compute the following, originally done in six-digit arithmetic, but using both three-digit round-to-even and three-digit truncation. **Assume that each constant and the result of each arithmetic operation must be converted to a machine number** using the appropriate float (fl) conversion.

$$\frac{17.4500 - 2.23606}{3.14159 - 2.71828} \; (= 35.9404)$$

Compute differences first, then quotient, then errors (and relative errors) and put them in the following table (please show your work below the table):

Method	Numerator	Denominator	Approximation	Error	Relative Error
3-digit truncation	15.1	.430	35.1	-.8404	2.34%
3-digit rounding	15.2	.420	36.2	.2596	.722%

- (10 pts) truncation:

Num: $17.4 - 2.23 = 15.17 \approx 15.1$

Den: $3.14 - 2.71 = .43 \approx .430$

Approx: $\frac{15.1}{.430} = 35.116 \approx 35.1$

Error $35.1 - 35.9404 = -.8404$

Rel Error $= \frac{.8404}{35.9404} = .02338$

- (10 pts) rounding:

Num: $17.4 - 2.24 = 15.16 \approx 15.2$

Den: $3.14 - 2.72 = .42 \approx .420$

Approx: $\frac{15.2}{.42} = 36.1904$

≈ 36.2

Error $36.2 - 35.9404 = .2596$

Rel Error: $\frac{.2596}{35.9404} = .00722$

20

Problem 2. (20 pts) Avoidable computational errors

a. (10 pts) We all know that the roots of a quadratic are given as $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$. However we shouldn't necessarily compute them that way. Compute the roots of the quadratic $x^2+500x+1$ using four-digit arithmetic. Do it in such a way that you obtain the best two roots (or approximations to the roots) that you can.

Root 1	-0.002
Root 2	-500

$x^2+500x+1$ ← b is positive. ∴ want $-b-$ same sign.

$$x_1 = \frac{2c}{-b-\sqrt{b^2-4ac}} = \frac{2(1)}{-500-\sqrt{500^2-4(1)(1)}} = \frac{2}{-500-\sqrt{250,000-4}}$$

250,000 4-digit ✓

$$= \frac{2}{-500-\sqrt{250,000}} = \frac{2}{-1000} = -0.002$$

249,996 round → 250,000

$$x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a} = \frac{-500-\sqrt{250,000-4}}{2(1)} = \frac{-500-\sqrt{250,000}}{2}$$

249,996 rounded

$$= \frac{-1000}{2} = -500$$ ✓

b. (10 pts) For which values of x do tiny relative errors in x result in large relative errors in $\cos(x)$?

Condition # of $\cos x$ → $\left|\frac{x\sin(x)}{\cos(x)}\right| = |x\tan x|$ as $E_x \to 0$

$x\tan x$ blows up around $\pi/2 + \pi k$ all solns ?

So small relative errors ($E_x \ll 1$) produce large relative errors near

$x = \pi/2\, k$

$\pi/2 + \pi k$, $n \in \mathbb{Z}$

19

Problem 3. (20 pts) Given the toy computer

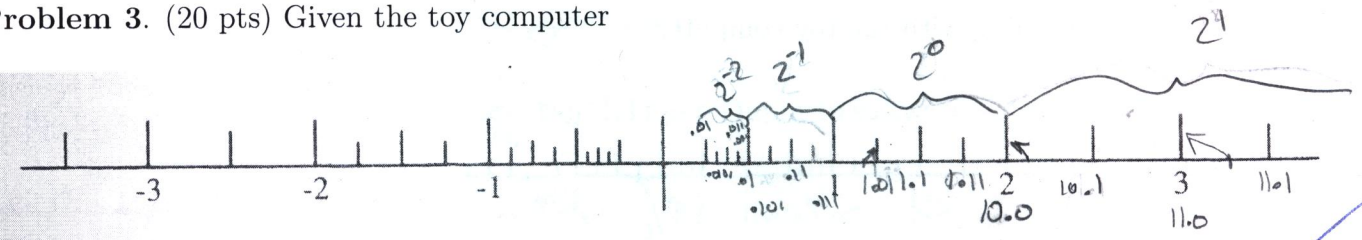


Figure 2.3: Machine numbers on a toy binary computer

a. (5 pts) Give the base, precision, and the exponent range (permissible values of e, the exponent) of the computer (you may simply describe it the same way that the authors did).

$\pm 2^e \pm b_1.b_2b_3$ $\quad e \in [-2, 1]$ $\quad b_1 = 1$

Base: 2
Precision: 3
Exponent Range: $[-2, 1]$

b. (5 pts) Assume round-to-even. What machine number will represent 1.2? 2.1? Compute $fl(fl(2.1) + fl(1.2))$. Illustrate this calculation on the diagram of the machine above.

x	Machine Number
1.2	$1.25_{10} = 1.01$
2.1	$2_{10} = 10_2$
$fl(fl(2.1) + fl(1.2))$	$3_{10} = 11_2$

$1.01 + 10$

$fl(11.0) = 11.0$

c. (10 pts) Compute the product $fl(fl(1.2) \cdot fl(0.43))$ using this toy computer, assuming rounding-to-even arithmetic.

x	Machine Number
$fl(fl(2.1) \cdot fl(1.2))$	$.1_2 = .5_{10}$

1.2 0.43

.25
.375
.4375
.5
.625

$(1.01) \cdot (.0111)$

$1.25 \cdot .4375 = .546875 \rightarrow .5$

Problem 3. (20 pts) Given the toy computer

Figure 2.3: Machine numbers on a toy binary computer

a. (5 pts) Give the base, precision, and the exponent range (permissible values of e, the exponent) of the computer (you may simply describe it the same way that the authors did).

$\pm\ 2^e \times b_1.b_2b_3 \quad -2 \le e \le 1$

b. (5 pts) Assume round-to-even. What machine number will represent 1.2? 2.1? Compute $fl(fl(2.1) + fl(1.2))$. Illustrate this calculation on the diagram of the machine above.

x	Machine Number
1.2	$2^0 \times 1.01$
2.1	$2^1 \times 1.00$
$fl(fl(2.1) + fl(1.2))$	$2^1 \times 1.10$

$$\begin{array}{r} .101 \times 2^1 \\ +\ 1.00 \times 2^1 \\ \hline 1.101 \times 2^1 \end{array} \approx 1.10 \times 2^1$$

c. (10 pts) Compute the product $fl(fl(1.2) \cdot fl(0.43))$ using this toy computer, assuming rounding-to-even arithmetic.

x	Machine Number
~~$fl(fl(2.1) \cdot fl(1.2))$~~	$2^{-1} \times 1.00$

$fl(fl(1.2) \cdot fl(0.43))$

$$\begin{aligned} &(2^0 \times 1.01)(2^{-2} \times 1.11) \\ &= (2^0 \times 1.01)(2^0 \times .0111) \\ &= 2^0 \times 0.10 \\ &= 2^{-1} \times 1.00 \end{aligned}$$

Problem 4. (20 pts) Continuing with the toy computer

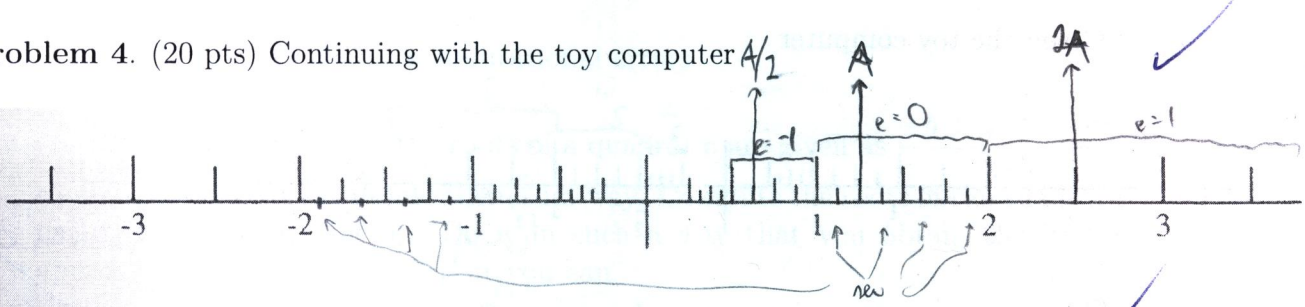


Figure 2.3: Machine numbers on a toy binary computer

$\pm 2^e(1.b_1b_2)$

a. (5 pts) Suppose we extend the possible exponents, without making any other changes to the machine. What machine numbers (answers given in base 10) would be added to the machine at the $e = 8$ level?

$\pm 2^8(1.00)_2 = \pm 2^8(1) = \pm 256$
$\pm 2^8(1.01) = \pm 2^8(1.25) = \pm 320$
$\pm 2^8(1.10)_2 = \pm 2^8(1.5) = \pm 384$
$\pm 2^8(1.11)_2 = \pm 2^8(1.75) = \pm 448$

b. (5 pts) If we added just one bit to the machine, without making any other changes to the machine, indicate clearly on the diagram above the numbers that would be added at the $e = 0$ level? How many new numbers would be added, and what are their values or machine representations?

8 new numbers added
$\pm 2^0(1.001) = \pm(1 + 1/8)$
$\pm 2^0(1.011) = \pm(1 + 1/4 + 1/8)$
$\pm 2^0(1.101) = \pm(1 + 1/2 + 1/8)$
$\pm 2^0(1.111) = \pm(1 + 1/2 + 1/4 + 1/8)$

good!

c. (10 pts) Clearly mark on the diagram above the machine number $(1.01)_2$ (call it A). Multiply A by two, and indicate the resulting number. Divide A by 2 and indicate that number. What do you notice about these operations and the positions of $2A$ and $A/2$ relative to the location of A?

2^0 $1 + 1/4$

$fl(2A) = fl(2.5)$

multiply and dividing by the base number is essentially changing your view point/exponent.
A is the second machine number at e=0, 2A is the second machine number at e=1.
and A/2 is the second machine number at e=-1

Well done

20

Problem 5: (20 pts) Short answer (5 points each):

a. Why do the authors advise using $4 \cdot \tan^{-1}(1)$ as an input to software, rather than any particular decimal approximation to π?

By inputting this to the system, the computer can easily save the #s 4 and 1, since those are base 2. Then it can compute the value of π to the highest accuracy it can achieve. If a decimal approximation of π is used, it will be coerced to the nearest binary which may have even more error than your decimal approximation.

b. Write 935_{10} in base 2

1110100111_2

$935 - 512 \Rightarrow 423 - 256 \Rightarrow 167 - 128 \Rightarrow 39 - 32 \Rightarrow 7 - 4 \Rightarrow 3 - 2 \Rightarrow 1$

$\frac{1}{2^9} \frac{1}{2^8} \frac{1}{2^7} \frac{0}{2^6} \frac{1}{2^5} \frac{0}{2^4} \frac{0}{2^3} \frac{1}{2^2} \frac{1}{2^1} \frac{1}{2^0}$

$2^9 = 512$, $2^8 = 256$, $2^7 = 128$, $2^5 = 32$

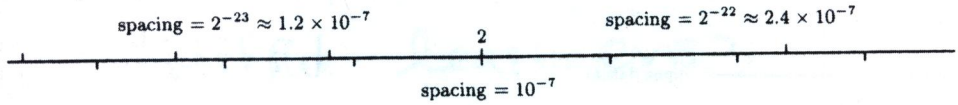
c. Write 10011011_2 in base 10.

155_{10}

$\frac{1}{2^7} \frac{0}{2^6} \frac{0}{2^5} \frac{1}{2^4} \frac{1}{2^3} \frac{0}{2^2} \frac{1}{2^1} \frac{1}{2^0}$

$128 + 16 + 8 + 2 + 1 = 155$

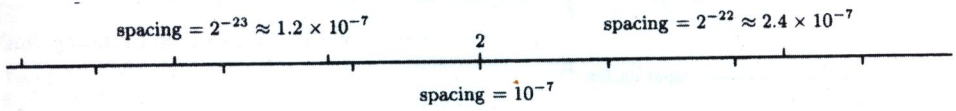
d. Consider the Figure on page 55, representing two computers that use IEEE single precision machine numbers above and 8-digit numbers below the axis: Why does the spacing between



the base 2 numbers increase to the right of 2, but the spacing of the base 10 machine does not?

2 is a power of 2. Therefore, the base 2 #s increase in exponent to the right (from 2^0 to 2^1), with each shift in exponent range, the spacing changes. The base 10 #s, however, remain in the same exponent range (10^0) since that range includes $[1, 10)$

d. Consider the Figure on page 55, representing two computers that use IEEE single precision machine numbers above and 8-digit numbers below the axis: Why does the spacing between



the base 2 numbers increase to the right of 2, but the spacing of the base 10 machine does not?

In base 2, you are experiencing an order change at 2 ($2^0 \rightarrow 2^1$), and thus your spacing between machine numbers will double. In base 10, you remain in the same order, so spacing remains the same. good

18.5

Problem 5: (20 pts) Short answer (5 points each):

a. Why do the authors advise using $4 \cdot \tan^{-1}(1)$ as an input to software, rather than any particular decimal approximation to π?

because π is non terminating so storing it's exact value would take an infinite amount of memory while an equation to represent π is feasible to store in a computer

Problem 6: (20 pts) Recall that the Babylonian algorithm for calculating $\sqrt{c}$ can be expressed as an iterative method, where we start with an initial value $x = x_0$, then

$$y = \frac{c}{x} \text{ and } x = \frac{x+y}{2}$$

Now do it again.

a. (10 pts) Assume that $x_0 = 1$, and use three-digit round-to-even arithmetic on all real numbers. Compute an approximation to $\sqrt{\pi}$ using two iterations of the Babylonian algorithm to compute x_1 and x_2. Assume that the true value is 1.77245. How well does the algorithm do?

$X_0 = 1$ $\quad y_0 = \frac{\pi}{1}$ $\quad X_1 = \frac{1+\pi}{2} = \frac{1+3.14}{2} = 2.07$

$X_1 = 2.07$ $\quad y_1 = \frac{\pi}{2.07} = 1.52$ $\quad X_2 = \frac{2.07 + 1.52}{2} = 1.795 =$ 1.80 = $X_2 \approx \sqrt{\pi}$

Relative Error = $\frac{1.80 - 1.77245}{1.77245} = 1.55\%$

The Algorithm does pretty well for only 2 iterations

good + 3 d.s.f arithmetic

b. (10 pts) Now express the total error in the result $f(\pi)$ as a sum of propogation data error and computational error. Use the ideas of Figure 1.8, and consider that **two** applications of the Babylonian algorithm (starting from $x_0 = 1$) is the approximation ($\hat{f}$) to the square root function (f). The data is $c = \pi$, but we have approximated it using three digits.

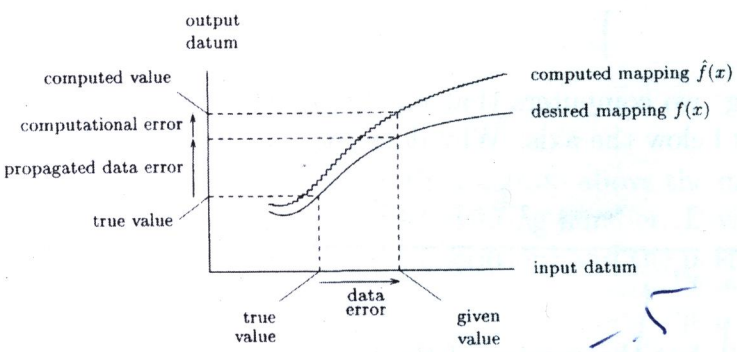


Figure 1.8: Data error vs. computational error

f = Sqrt Function

propagated error = $f(\hat{a}) - f(a)$

Computational error = $\hat{f}(\hat{a}) - f(\hat{a})$

$a = \pi$ $\quad \hat{a} = 3.14$

$f(\hat{a}) = \sqrt{3.14} = 1.77$

propagated = $|1.77245 - 1.77| = .00245$

Computed = $1.80 - 1.77 \approx .03$

Total Error = $.03 + .00245 = .03245$

$\hat{f}(\hat{a}) - f(\hat{a}) + f(\hat{a}) - f(a)$

$= .0279954$

19.5

Problem 7: (20 pts) Consider Example 2.7, p. 67:

"The calculation of $x - \pi$ is typical of what must be done to compute trigonometric functions. Assume the use of four-decimal-digit numbers. If you use $x - 3.142$, you will get a completely wrong answer for $x = 3.142$ and poor accuracy for nearby values. However, $f(x) = (x - 3.141) - 0.0005927$ produces results correctly rounded to four significant digits for all x."

a. (10 pts) Calculate the correct values of the following, using four-decimal-digit arithmetic and round-to-even: fl(-0.041 - 0.0005927) - fl(-0.04159) = -0.04159

$f(3.000)$	$f(3.100)$	$f(3.140)$	$f(3.142)$	$f(\pi)$
-0.1416	-0.04159	-0.001593	0.0004073	0.0004073

fl(0.001 - 0.0005927) = 0.0004073

fl(fl(3 - 3.141) - 0.0005927) = fl(-0.141 - 0.0005927) = fl(-0.14159927) = -0.1416

fl(fl(-0.001 - 0.0005927)) = fl(-0.0015927) = -0.001593

fl(0.001 - 0.0005927) = fl(0.0004073) = 0.0004073

-.5

b. (5 pts) Explain the significance (and even the cleverness) of this idea, and why it works.

The first subtraction deals with numbers on similar scales, resulting in a number similar in scale to the second subtraction. This insures that small changes in x will be "heard" or "felt" in f(x), instead of just being truncated.

Nicely stated.

c. (5 pts) The authors conclude that "...a carefully crafted expression like this can be undone by an optimizing compiler that assumes that the associative law holds for floating-point addition." What do they mean to suggest that the compiler might do?

The compiler might assume that $[(x - 3.141) - 0.0005927] \equiv [x - (3.141 + 0.0005927)]$ which is <u>not</u> true in operations involving machine numbers. Addition/subtraction is not associative or commutative for machine numbers.

Well done

-.5 • Watch that equal!

Re compiler writers 3.142...

c. (5 pts) The authors conclude that "...a carefully crafted expression like this can be undone by an optimizing compiler that assumes that the associative law holds for floating-point addition." What do they mean to suggest that the compiler might do?

If we code this function into a compute with an optimizing compiler, the compute might use the associative function and change it to: $f(x) = x - (3.141 + 0.000592)$. This will cause the function to become: $f(x) = x - 3.142$ which brings us right back to the original problem. A way around this could be to use two functions:

$f(x) = x - 3.141$

$g(y) = f(y) - 0.000592$

✓ good.