CS100 B
Programming Assignment 3

Due: Thursday, October 14

1. Goals

This assignment will help you develop skills in ***object oriented programming*** (OOP) by using methods in Java. You will implement an approach of applied mathematics called ***Interval Analysis*** (IA) which, in turn, incorporates principles of Computer Science and Discrete Mathematics. Applying IA with OOP will help you solve a circuit design problem from Electrical Engineering. Yes, problem solving with programming tends to cover all kinds of fields of study!

2. Motivation

Problem solving is fundamental in all engineering and science. To solve a problem, one usually uses experiments, theory, computation, or combinations thereof. For example, chemistry experiments help discover how various chemicals will react under a variety of conditions. Computation also helps resolve uncertainty by testing new algorithms and finding answers to complicated systems. All approaches confront some aspect of ***uncertainty***. Uncertainty means many things to many fields of study, such as errors, imprecision, or just lack of knowledge.

Uncertainty in the "real world" drives researchers to constantly improve models, and hopefully, improve accuracy. But, an interesting dilemma arises. Whereas models strive for accurate representations of real-world systems, real-world information lacks precision or eludes discovery. Consider the material and geometric properties of structures that surround you. Although the designers have specified certain properties and dimensions, the true values actually vary, sometimes over a wide range of values. Usually designers try to bound these errors with ***tolerances***, the range of over which the model can "accept." That is, a designed structure will not fail given deviations in the assumed or measured design parameters.

Have you ever wondered why some companies so rigorously test devices after building them? As discussed above, models cannot account for all uncertainty. Extensive experimentation may still not account for unforeseen behaviors. Also, not all systems, such as permanent structures or limited-production devices, can afford rigorous testing. However, some researchers have suggested predicting uncertainty in a real-world system by incorporating the same uncertainty in the modeling and design process.

Typically, probability and statistics manage uncertainty of a model. A designer tries to design against failure for a statistical range of data. Unfortunately, statistical methods lack usefulness for scare data. Other approaches offer alternatives to the "standard" probabilistic approach. Rather than solving for what is *probable*, some approaches determine what is *possible* over a range of design parameters. This assignment demonstrates one such technique called Interval Analysis.

3. Interval Analysis

Interval Analysis (IA) uses intervals to represent uncertainty in a model to capture a model's range of behaviors using set-based mathematics. This section reviews the fundamental rules and properties of the approach.

3.1 Notation

This section briefly reviews mathematical notation used throughout this assignment:

- $\{a, b, c\}$: a set containing elements a, b, and c
- $a \in x$: a belongs to the set x
- $a \subset b$: a is a subset of b; also, b contains a
- $a \subseteq b$: $a \subset b$, but $a = b$ might also hold
- $[a, b]$: the set of all numbers between a and b, also called an interval
- $\{x|y\}$: the set of all x such that the condition y holds
- $a \equiv b$: a and b are equivalent statements

3.2 Background

Moore (1966) originally applied IA to contain round-off error in computer arithmetic. For instance, consider the decimal equivalent of $\frac{1}{3}$:

$$\frac{1}{3} = 0.333\ldots. \qquad \textbf{(1)}$$

Applications that require the utmost in precision will encounter round-off error. Instead, a program might use an interval to contain the exact quantity. For example,

$$[0.3333, 0.3334] \qquad \textbf{(2)}$$

which contains $\frac{1}{3}$. Equation 2 guarantees containment of the correct result regardless of round-off error. Other values, like irrational numbers, could benefit from such treatment.

3.3 Interval Definition

Assume a quantity x is uncertain and can, thus, vary between two given values, a lower bound $\underline{x}$ and an upper bound $\bar{x}$. You c an think of this range of values with inequalities:

$$\underline{x} \le x \le \bar{x}. \qquad \textbf{(3)}$$

Assuming all real values, an interval $\boldsymbol{x}$ represents the range of all values $\underline{x} \le x \le \bar{x}$. Typically, square brackets denote an interval as

$$\boldsymbol{x} = [\underline{x}, \bar{x}] = \{x|\underline{x} \le x \le \bar{x}\}, \qquad \textbf{(4)}$$

where $\{x | \underline{x} \le x \le \bar{x}\}$ means “the set of all values of x such that $\underline{x} \le x \le \bar{x}$.” Boldface notation $\boldsymbol{x}$ denotes an interval quantity.

The interval $\boldsymbol{x}$ stores the set of all values of x between, and including, the lower and upper bounds $\underline{x}$ and $\bar{x}$. For instance, the interval $[1, 2]$ contains the set of all numbers between lower bound 1 and upper bound 2:

$$\boldsymbol{x} = [1, 2] \equiv 1 \le x \le 2 . \tag{5}$$

3.4 Width Models Uncertainty

Width $\mathrm{w}(\boldsymbol{x})$ measures the “thickness” of an interval $\boldsymbol{x}$:

$$\mathrm{w}(\boldsymbol{x}) \equiv \bar{x} - \underline{x} . \tag{6}$$

Do you see a connection between interval width and uncertainty? An interval can model a parameter’s range of values, assuming lower and upper bounds. What happens if a quantity is complete and unequivocally certain? Then, that quantity has zero width and is called ***crisp***.

3.5 Tolerance

IA helps model tolerances by assuming interval width represents the range of uncertainty of a parameter. Typically, designers choose tolerances based on percentages of assumed values. For instance, a beam designed to be 3 m in length might vary by $\pm 1\%$, either shorter or longer. In terms of an interval, beam length $\boldsymbol{L}$ would, then, vary as follows:

$$\boldsymbol{L} = [3 - (3)(1\%), 3 + (3)(1\%)] . \tag{7}$$

In general, express ***tolerance*** as a percentage of a particular quantity. But, *which* quantity?

IA hasn’t defined specific rules for choosing the “initial” value. For now, assume a model produces an ideal value that a designer uses. This value, called the center value x^c, forms the center of an interval $\boldsymbol{x}$. Borrowing from probability, you can also think of the center value as a kind of “average” quantity.

Let p represent a ***percent uncertainty*** of a parameter in a model. Multiply p by the parameter’s center value to produce a tolerance which can be added or subtracted. In general, let

$$0 \le p \le 1 , \tag{8}$$

where

- $p = 0$ corresponds to 0% uncertainty, or a certain value.
- $p = 1$ corresponds to 100% uncertainty, or a very uncertain value.

For instance, $p = 0.01$ means 1% uncertainty.

Using percent uncertainty to add or subtract from a center value models a tolerance problem. Thus,

$$\underline{x} = x^c - px^c \text{ and } \bar{x} = x^c + px^c , \tag{9}$$

as demonstrated in Equation 7. Substituting values from Equation 9 into Equation 4 produces the interval

$$\boldsymbol{x} = (x^c)[1 - p, 1 + p], \quad \textbf{(10)}$$

which models the tolerance for an uncertain parameter. Figure 1 depicts an interval $\boldsymbol{x}$ as a range of these values.

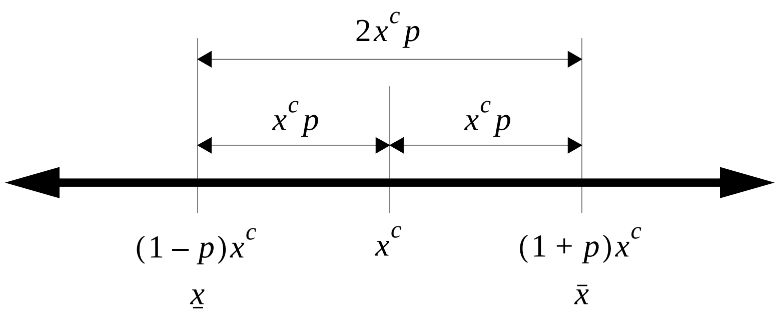

Figure 1: Interval about Center Value

3.6 Creating Intervals

To create an interval, follow these steps:

- Assume/obtain/use a center value x^c.
- Assume/obtain/use a percent uncertainty p, where $0 \le p \le 1$
- Compute $(x^c)[1 - p, 1 + p]$.

If you prefer to use lower bound $\underline{x}$ and upper bound $\bar{x}$, use the following formulas, derived from Equation 10:

$$\bar{x} = x^c(1 + p) \quad \textbf{(11)}$$

and

$$\underline{x} = x^c(1 - p). \quad \textbf{(12)}$$

3.7 Arithmetic Operations

So, how does someone manipulate and combine intervals when performing mathematical operations? IA employs two principles to guarantee containment of all possible values:

1. ***Independence***: Numerical values vary independently between intervals.
2. ***Extremes***: Interval operations should produce the largest possible outer bounds.

An interval arithmetic operation generates the widest possible bounds using any value inside the intervals. Consult Table 1 for rules of interval arithmetic, given two intervals $\boldsymbol{x}$ and $\boldsymbol{y}$. For example, given the equation $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{y}$ with $\boldsymbol{x} = [1, 2]$ and $\boldsymbol{y} = [3, 4]$, solve for $\boldsymbol{z}$:

$$\boldsymbol{z} = [1, 2] + [3, 4] = [4, 6]. \quad \textbf{(13)}$$

Note that careful consideration of the signs of both intervals produces more efficient rules for multiplication and division.

Table 1: Interval Arithmetic Rules

Operation		Rule
addition	$\boldsymbol{x}+\boldsymbol{y}$	$[\underline{x}+\underline{y}, \bar{x}+\bar{y}]$
subtraction	$\boldsymbol{x}-\boldsymbol{y}$	$[\underline{x}-\bar{y}, \bar{x}-\underline{y}]$
multiplication	$\boldsymbol{xy}$	$[\min(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}), \max(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y})]$
division	$\boldsymbol{x}/\boldsymbol{y}$	$[\min(\underline{x}/\underline{y}, \underline{x}/\bar{y}, \bar{x}/\underline{y}, \bar{x}/\bar{y}), \max(\underline{x}/\underline{y}, \underline{x}/\bar{y}, \bar{x}/\underline{y}, \bar{x}/\bar{y})], 0 \notin \boldsymbol{y}$

3.8 Arithmetic Properties

Table 2 summarizes and demonstrates further properties of interval arithmetic. Due to interval independence, distributive and cancellation properties of standard arithmetic do not hold. Different orders of operation might yield different results. Therefore, attempting an inverse operation, as in crisp arithmetic, cannot retrieve a “previous” value. For example,

$$[1, 2][3, 4] = [3, 8]. \tag{14}$$

But, attempting to reverse the operation in Equation 14 fails:

$$[3, 8]/[3, 4] = \left[\frac{3}{4}, \frac{8}{3}\right] \supset [1, 2]. \tag{15}$$

Table 2: Interval Arithmetic Properties

Property	Addition and Subtraction	Multiplication and Division
commutative	$\boldsymbol{x}+\boldsymbol{y} = \boldsymbol{y}+\boldsymbol{x}$	$\boldsymbol{xy} = \boldsymbol{yx}$
associative	$\boldsymbol{x}+(\boldsymbol{y}+\boldsymbol{z}) = (\boldsymbol{x}+\boldsymbol{y})+\boldsymbol{z}$	$\boldsymbol{x}(\boldsymbol{yz}) = (\boldsymbol{xy})\boldsymbol{z}$
neutral/identity elements	$0+\boldsymbol{x} = \boldsymbol{x}+0 = \boldsymbol{x}$	$(1)\boldsymbol{y} = \boldsymbol{y}(1) = \boldsymbol{y}$
subdistributivity	$\boldsymbol{x}(\boldsymbol{y} \pm \boldsymbol{z}) \subseteq \boldsymbol{xy} \pm \boldsymbol{xz}$ (equality holds for crisp $\boldsymbol{x}$)	
subcancellation	$\boldsymbol{x}-\boldsymbol{y} \subseteq (\boldsymbol{x}+\boldsymbol{z})-(\boldsymbol{y}+\boldsymbol{z})$	$\boldsymbol{x}/\boldsymbol{y} \subseteq (\boldsymbol{xz})/(\boldsymbol{yz})$
	$0 \in \boldsymbol{x}-\boldsymbol{x}$	$1 \in \boldsymbol{a}/\boldsymbol{a}$

4. Application of IA

This section applies to IA to a small circuit design problem.

4.1 Amplifier

Figure 2 illustrates a model of an amplifier circuit. In general, an ***amplifier*** produces an output signal from an input signal. An amplifier will reduce or increase the magnitude of the input signal, which includes voltage V or current i. Amplifier gain G is defined as the ratio of voltage output and voltage input. Thus, $G = V_2/V_1$. A is called the operational amplifier gain, which represents the factor that modifies the current exiting the amplifier. From the theory of circuits, the following equation provides G:

$$G = \frac{1}{\frac{1}{A} + \frac{R_1}{R_1 + R_2}}, \tag{16}$$

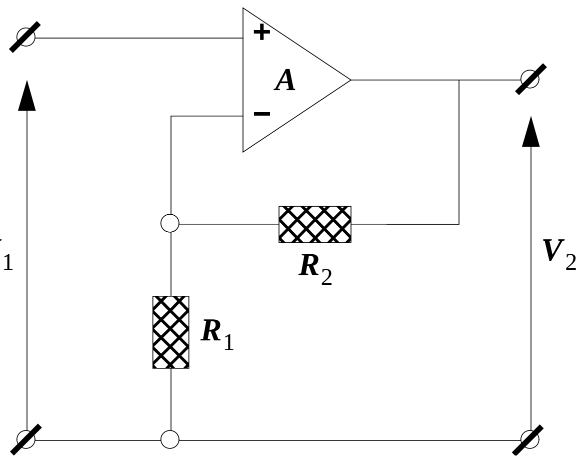

Figure 2: Amplifier Circuit (Kolev, 1993)

where R_1 and R_2 are values of resistance in Ohms (Ω).

4.2 Tolerance

Assume that a designer is looking for the worst-case tolerance limits of G. Also, let intervals bound the values of R_1, R_2, and A:

- Assume a percent uncertainty of 2% for R_1 and R_2.
- Assume a percent uncertainty of 10% for A.

4.3 Interval Operations

The IA method allows you to adapt standard equations, like Equation 16, with intervals. In general, for each parameter, substitute the tolerance interval given a center value. In this example, use the center values $R_1^c = 100\Omega$, $R_2^c = 900\Omega$, and $A^c = 100$. For instance, wherever you see R_1, you would use $\boldsymbol{R}_1 = [98, 102]\Omega$ with interval arithmetic operations instead of "normal" arithmetic, as demonstrated in Equation 13. Consequently, the equation for G is represented as

$$\boldsymbol{G} = \frac{1}{\frac{1}{\boldsymbol{A}} + \frac{\boldsymbol{R}_1}{\boldsymbol{R}_1 + \boldsymbol{R}_2}} \tag{17}$$

in interval notation. For instance, the expression $\boldsymbol{R}_1 + \boldsymbol{R}_2$ yields

$$[98, 102] + [882, 918] = [980, 1020]. \tag{18}$$

You may wish to manipulate terms algebraically to improve the interval containment. For instance, the alternative form

$$\frac{1}{1 + \frac{R_2}{R_1}} = \frac{R_1}{R_1 + R_2} \tag{19}$$

produces a "tighter" interval due to no repetition of the R_1 term.

5. Problems

Find the worst-case enclosure of G given the data in Section 4 for the model in Figure 2. Perform the following steps to produce your result.

5.1 Manual

Never completely trust software! All results of computation should be checked by an independent source, if possible. For this assignment, the model is small enough to check by hand. On a separate sheet of paper, solve Equation 17 "by hand."

- Be sure to title this sheet **Manual Results** and show all data and steps.
- Feel free to use a calculator, but you must type your results.
- Clearly indicate the final result for $\boldsymbol{G}$.

5.2 Programming IA

Write a program that performs interval arithmetic between two intervals $\boldsymbol{x}$ and $\boldsymbol{y}$. Be sure to account for the following:

- The user must enter the center and percent uncertainty values for intervals $\boldsymbol{x}$ and $\boldsymbol{y}$.
- Use a class called `Interval` that includes methods for interval arithmetic operations: addition, subtraction, multiplication, and division.
- Be sure to account for users that might enter a crisp number ($p = 0$).
- Test the following examples by hand and using your code and produce output for each:

  5.2a) $[1, 2] + [3, 4]$

  5.2b) $[1, 2] - [1, 2]$ (result is not zero)

  5.2c) $[1, 2] \times 2$

  You do not need to report your "manual" answers.

5.3 Solving Tolerance Problem

You will now adapt your program to solve the tolerance problem discussed previously:

- Write a method called `amplifier` inside the class that contains `main`. If you use CodeWarrior, place `amplifier` inside the `CUCSApplication` class.
- `amplifier` should declare values for all interval data shown Section 4.
- `amplifier` should compute and output the interval $\boldsymbol{G}$.

You must use `Format.java` to format your output. Assume that your output should have at least two significant figures.

5.4 Comparison

How well does IA perform a worst-case tolerance problem? Write another program using new stationery to solve for $\boldsymbol{G}$ the "long way." What is this longer method? Since there are only three parameters bounded by two values, there are a total of 8 possible ways to pick a combination of the outer bounds. Write a program that solves all 8 cases and reports $\boldsymbol{G}$ without IA. You can also add a method to your other program (Section 5.3) if you prefer. Hints:

- Think of nested loops. Each parameter can increment from lower to upper value using the width of its interval.
- At the "innermost" position of the nested loop, you can compute G using crisp values of all parameters, since the loops have "chosen" the values for you.
- When you compute G, you should check whether the current value is a maximum or minimum value. As you check all 8 values, doing a max/min check will eventually give you the outer bounds.
- Those outer bounds form the actual, or ***true***, worst-case tolerance interval.

5.5 Discussion

After developing, testing, and running your code, answer these questions. Please type all answers on a separate sheet:

1. Write a title, **Answers to Discussion Questions**.
2. How does IA result compare with the true result for computing $\boldsymbol{G}$?
3. Consider the example $[1, 2] - [1, 2]$. Does your program give you the same answer if you compute this example by hand? Why does this example NOT equal zero? (Hint: What principles underlie IA?)
4. Based on *your* results, which approach, IA or true, is more efficient? Why?
5. Based on *your* results, which approach, IA or true, is more accurate? Why?
6. Can you think of a way of improving IA's performance?

6. Submitting Your Work

6.1 Due Date

This assignment is due in lecture on Thursday, October 14, 1999. You may turn it in to a consultant before that date in the consulting room in Carpenter. Do not turn it in at Carpenter on the due date. Late programs will not be accepted.

6.2 Labeling Your Work

Always write your name, Cornell ID#, and the day/time/instructor for your section in the first comment of each program you hand in for credit. Otherwise, the program will not be graded. All solutions and commentary must be typed! If you wrote the program with a partner, turn in only one printout with your partner's name and ID# in the comment, as well as your own. The comment must also include the section day/time/instructor for the partner. The program will be returned to the first person listed. Sign your name(s) in the comment. Please staple the pages of your assignment together.

6.3 Grading

This assignment will be given two grades: the first based on correctness, the second on program organization and style. Each grade will be a 0, 1, or 2. Not only should your program work, but it should contain adequate comments to guide the reader who is interested in understanding it. The declaration of every significant variable should include a comment describing that variable. There should be appropriate comments in the code so the reader can see the structure of the program, but not so many that the program text is hard to read.

6.4 What to Hand In

Read Section 4 (Programs) of the *CS100 General Information* packet for explanations of requirements. For this assignment, staple the following sheets together:

- Cover sheet with your name(s), student ID(s), and the title PROGRAMMING ASSIGNMENT 3.
- The results of Section 5.1. Don't forget to title the sheet.
- The program(s) for Sections 5.2, 5.3, and 5.4. If you use the same code over, you do not need to print it twice, so long as your code can be cleared followed.
- Output of Section 5.2 for the three test cases. You can include all three outputs on one sheet.
- Output of Sections 5.3 and 5.4. Since these results use different programs, output should be on different sheets. Be sure to label your output using print statements in your code!
- Typed answers to questions in Section 5.5. Don't forget to title this sheet.

7. References

If you are interested in learning more about Interval Analysis, check out

- http://cs.utep.edu/interval-comp/main.html
- http://cs.utep.edu/interval-comp/appl.html
- Kolev, L. V. (1993). *Interval Methods for Circuit Analysis*, World Scientific, New Jersey.
- Moore, R. E. (1966). *Interval Analysis*, Prentice Hall, New Jersey.
- Schwartz, D. I. (1999). *Deterministic Interval Uncertainty Methods for Structural Analysis*, Ph.D. Dissertation, State University of New York at Buffalo.

For more information about circuit design and Electrical Engineering, check out

- http://www.ee.cornell.edu/
- http://www.ieee.org/