Computer Application: Area Under the Curve Probabilistic Approach

Charles Binder

Pascack Valley High School
Hillsdale, New Jersey

All of us are familiar with the typical calculus application of determining the "area under the curve." We teach this concept in our calculus classes with appropriate integration techniques, and when we cannot integrate the function in question, we revert to finding the approximate area for the interval. The most frequently used techniques are: rectangular approximations, the Trapezoidal Rule, and if, sophistication permits, Simpson's Rule.

These approximation techniques can be readily adapted to computer analysis; however, the implementation of any of them requires a rather thorough knowledge of the underlying concepts, and usually a good understanding of the limit concept.

The probabilistic approach described here will work on most functions, and requires a lesser degree of sophistication in both the programming and calculus concepts. It can best be described by making an analogy to a dart game and the dart board.

Consider a game of darts, where the dart board consists of two different colored regions. Your goal is to calculate the area of one region, knowing only the entire surface area of the board. The rules are simple:

every dart thrown will land on the board, and its position is determined purely by random chance. Consider an unlimited number of darts where we keep track of the count of those that fall into our "targeted" region.

After many darts have been tossed we can find the ratio of darts that fall into the "targeted" region in comparison to all those thrown. The product of this ratio with the area of the entire surface will serve as our approximation.
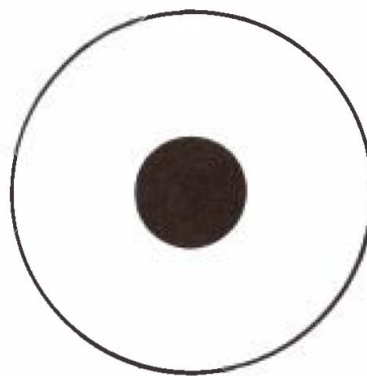
Suppose that the area of a dart board is 81π sq. in., and of 500 darts thrown, 100 land in the "targeted" region. Thus the area of this region can be approximated by:
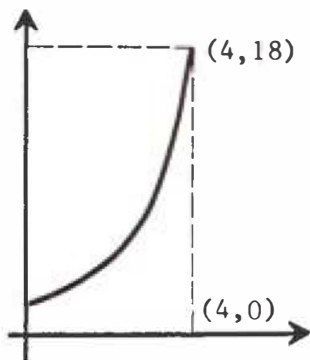
$$\frac{100}{500} * 81\pi = \frac{81\pi}{5}$$

Obviously, the more darts thrown, the better this approximation. This is where our tireless computer will come in handy.

The proper strategy for using a mathematics function and simulating this on a computer will be:

1. Sketch the curve in question, noting all boundaries.

2. Encompass this region within a rectangle of known dimensions.

3. Throw the darts and tally those that fall in the "region."

4. Compute the approximate area by taking the product of the ratio in no. 3 with the known area of the rectangle.

Now let's analyze these details with an example. Suppose our problem is to find the area of the region bounded by $y = x^2 + 2$, the x and y axes, and the line x = 4.
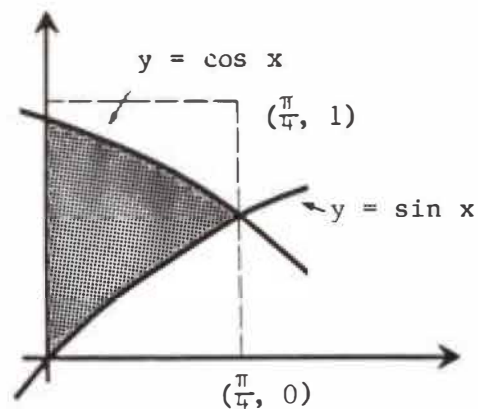
4. The tally will be accomplished by comparing the ordinate (Y) of our dart with the ordinate (for the same abscissa), on the curve:

IF Y <= X↑2 + 2 THEN T = T + 1.

If the ordinate for the dart is less than or equal to the calculated ordinate for the curve, then we will tally (T) this result. See listing no. 1 for the complete program to approximate this area.

This technique can be applied to areas of regions between two curves with only minor changes. Again, the sketch is very important, not only for the boundaries, but also to know which curve is "above" the other.

To illustrate this, let's take the area of the region bounded by $y = \sin x$ and $y = \cos x$, from 0 to $\pi/4$.

1. Sketch the curve, as shown, indicating boundary points.

2. This region can be inscribed in a rectangle whose vertices are (0,0), (4,0), (4,18), and (0,18).

3. Using the random number function with appropriate "stretching" (multiplying the random function by a value to keep the x and y values within the rectangle) parameters, we can generate the "location" of each dart by these statements: X = 4 * RND(0) and Y = 18 * RND(0).

1. This sketch would clearly show that y = cos x is "above" y = sin x in this interval.

2. The rectangle would be one with vertices at (0,0), ($\pi/4$,0), ($\pi/4$,1), and (0,1), which would have an area of $\pi/4$, or approximately 0.7853.

3. The "dart" would have coordinates of:
    X = 0.7853 * RND(0), Y = RND(0).

4. Our check to determine whether it is in the desired region would involve a compound statement:

   IF Y <= COS(X) AND Y >= SIN(X) THEN T = T + 1.

   (Refer to program listing no. 2 for the complete version of this problem.)

This probabilistic concept can easily be extended to stimulate students with other mathematical concepts. Two rather easy applications are approximating π and determining the value of ln x.

The approximation of π can be determined by attempting to find the area of a unit circle. Since $A = \pi r^2$, if the radius is 1, then the area must be π. First though, let's consider finding the area of a semicircular region, then doubling that.

The equation for the "upper" semicircle, $y = \sqrt{1 - x^2}$ can be used to easily determine the approximate area of the semicircular region. Please note that the x coordinate will have to be both "stretched and shifted" with our random function to generate the correct set of values (since the abscissas can be anywhere in the interval from −1 to 1).

X = 2 * RND(0) − 1   Y = RND (0)

Refer to listing no. 3 for the completed program.

The value for the ln x can be determined by referring to the definition that Thomas gives in his calculus texts: $\ln x = \int_1^x 1/t \, (dt)$, which is an area.

Consider the case where x is greater than 1; a program can be set up dependent on a value for x entered from the keyboard by an operator of the program. The appropriate rectangle to be used will be one whose dimensions are 1 (height) by x − 1 (width), and this can readily be determined from our value entered from the keyboard. Refer to program listing no. 4 for the complete solution.

The program listings given on page 35 are generic enough to run on almost any computer. Slight changes may be necessary if your syntax for the random number function differs. I have used the form of this function that generates the random value between 0 and 1. Since most BASIC implementations allow five significant digits, taking into account the typical "rounding" errors, the results generated here will be improved by using more "darts" and averaging answers found on successive runs.
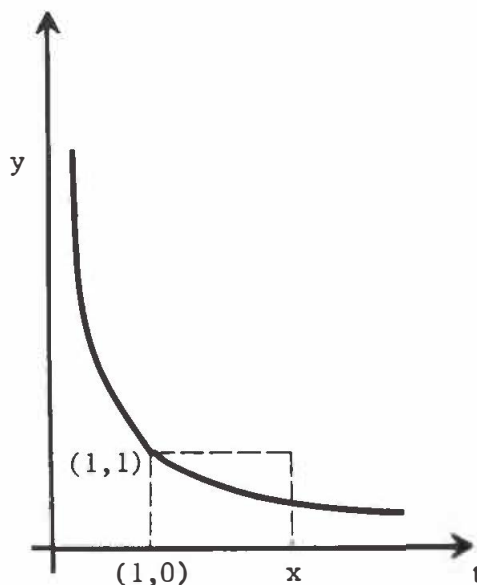
Probabilistic simulations can be exciting and can stimulate the students to pursue other areas. As you have seen, they do not require the student to have a sophisticated knowledge of either programming or calculus. Perhaps by examining these they will better appreciate not only the mathematical concepts, but will understand more fully how a computer can be a useful tool.

Reference

Thomas, George. Calculus and Analytical Geometry. Addison-Wesley Publishing Co., 1960.

Program Listings

```
5 REM LISTING # 1
10 REM APPROXIMATES AREA OF REGION BOUNDED BY
20 REM Y ± X↑2 + 2, THE X AND Y AXES, AND X = 4
30 T = 0
40 FOR I = 1 TO 500
50     X = 4*RND(0)
60     Y = 18*RND(0)
70     IF Y <= X↑2 + 2 THEN T = T + 1
80 NEXT I
90 A = (T/500)*72
100 PRINT"AREA IS APPROXIMATELY ";A
110 END


5 REM LISTING # 2
10 REM APPROXIMATES AREA OF REGION BETWEEN
20 REM Y = SIN X AND Y = COS X FROM 0 TO 1/4 PI
30 T = 0
40 FOR I = 1 TO 500
50     X = 0.7853*RND(0)
60     Y = RND(0)
70     IF Y <= COS(X) and Y >=SIN(X) THEN T = T + 1
80 NEXT I
90 A = (T/500)*0.7853
100 PRINT"AREA IS APPROXIMATELY ";A
110 END


5 REM LISTING # 3
10 REM APPROXIMATES PI BY COMPUTING THE
20 REM AREA OF A UNIT CIRCLE
30 T = 0
40 FOR I = 1 TO 500
50     X = 2*RND(0) -1
60     Y = RND(0)
70     IF Y<=SQR( 1 -( X↑2)) THEN   T = T + 1
80 NEXT I
90 A = (T/500)*2
100 P = 2 * A
110 PRINT"PI IS APPROXIMATELY ";P
120 END


5 REM LISTING # 4
10 REM APPROXIMATING THE VALUE OF LN X
20 REM X INPUT FROM KEYBOARD
30 PRINT"INPUT X"
40 INPUT X1
50 IF X1 <= 1 GOTO 30
60 FOR I = 1 TO 500
70     X = 1 + (X1 - 1)* RND(0)
80     Y = RND(0)
90     IF Y <= 1/X THEN T = T + 1
100 NEXT I
110 A = (T/500)*(X1-1)
120 PRINT"APPROXIMATE VALUE OF LN(";X1;") IS ";A
130 END
```