Fourth Programming Assignment – Due April 18 at 11:59 pm[1]

Objective and overview

This exercise should give you the ability to write code using the various repetition (looping) structures: (1) the do-while statement, (2) the do-loop-until statement, and (3) the for statement.

This exercise has four tasks. You can place the VBA code for all four tasks of this assignment in a single module, but you should use a separate worksheet for the first task. You can put the results of tasks two, three and four on a single worksheet.

Specific tasks

Task one: Nested loops – During class we discussed the use of nested loops to compute a table of kinetic energy. During various lectures and in-class exercises, we worked on generating the two-dimensional table and showed how to prepare row and column headers for mass and velocity. A full completed table of this type is obtained by using nested for loops and the cells method to write output to the Excel worksheet. A complete result for a table of kinetic energy would look like the one shown at the right.

Table of Kinetic Energies in Joules				
	V = 5 m/s	V = 10 m/s	V = 15 m/s	V =
m = 10 kg	125	500	1,125	
m = 20 kg	250	1,000	2,250	
m = 30 kg	375	1,500	3,375	
m = 40 kg	500	2,000	4,500	
m = 50 kg	625	2,500	5,625	
m = 60 kg	750	3,000	6,750	
m = 70 kg	875	3,500	7,875	
m = 80 kg	1,000	4,000	9,000	
m = 90 kg	1,125	4,500	10,125	
m = 100 kg	1,250	5,000	11,250	

In this task, you are asked to construct similar table that gives the required periodic loan payment, A, per dollar borrowed, P. This table will show the ratio A/P, as a function of the interest rate, i, and the number of required payments, n; this A/P ratio is given by the following formula.

$$\frac{A}{P} = \frac{i}{1-(1+i)^{-n}}$$

In this formula, i is the interest rate (for one payment period) and n is the number of payment periods. For example, many credit cards have a monthly interest rate of 1.5%. If you had this interest rate and wanted to pay off your entire credit card balance in 10 months (with no new charges) you could compute your monthly payment by first computing the A/P ratio for n = 10 and i = 1.5%. (Note that the interest rate of 1.5% is entered as the fraction 0.015 in this formula.)

$$\frac{A}{P} = \frac{i}{1-(1+i)^{-n}} = \frac{0.015}{1-(1+0.015)^{-10}} = 0.10843$$

If you make ten monthly payments of 0.10843 times your current balance, you would pay off the balance completely. *E.g.*, if your current balance is $1,000, you could pay it off with ten payments of $108.43.

[1] Late submissions accepted up to 11:59 pm, Friday, April 21 with 30% late penalty. No submissions accepted after this time.

In this task, you should create a table that shows the value of A/P as a function of n and i. (Conventionally, i is the row variable and n is the column variable in such tables.) Your table should have a format like the one for the kinetic energy shown on the previous page. Use interest rate values from 0.05% to 2% in increments of 0.05%. (Remember that the value of i in the formula is a fraction not a percent. Thus when i is 1.5%, 1 + i = 1.015.) Use values of n from 1 to 48 with an increment of 1. Use six decimal places for the output of A/P, which is usually a number less than one. Print the interest rate in percent and make sure that your code prints the final entry for i = 2%.

Tasks two to four: Summing infinite series – We have also discussed summing infinite series using different kinds of loops. In these tasks, you will use three kinds of loops to compute the cosine, using the infinite series for the cosine of an angle, x, measured in radians:

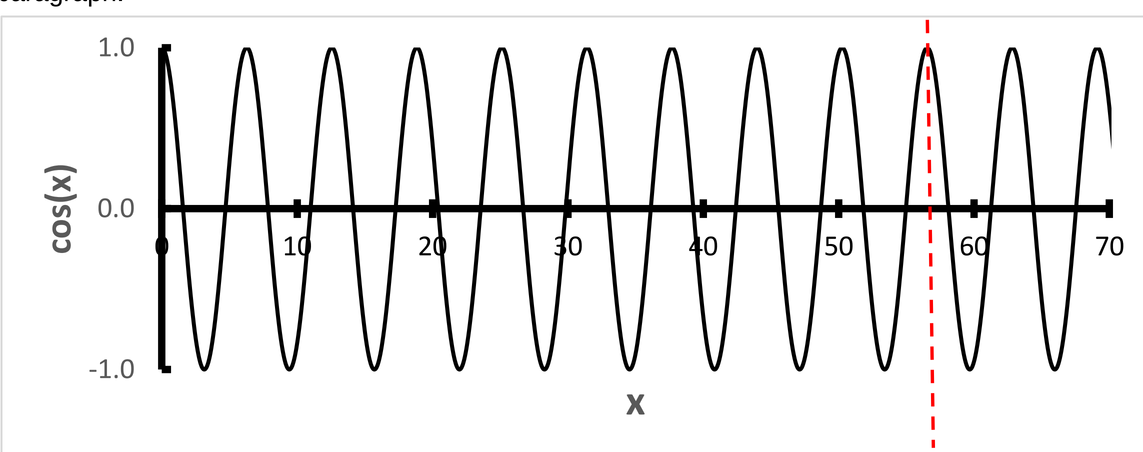
$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots$$

As we showed in class, this series sum can be analyzed as the sum of terms, T_n, where $T_0 = 1$ and subsequent terms can be found from the previous term by the general equation:

$$T_n = \frac{-x^2 T_{n-1}}{2n(2n-1)} \qquad n = 1, \ldots \infty$$

Specific assignment for the tasks two to four In these three tasks you will write three VBA functions to compute cos(x) using three different approaches. In task two, you will write the VBA code that uses a for loop. In task three you will replace the for-loop code with a do while loop that uses a Boolean variable, converged. You can use the example of the Do While loop for the computation of a square root shown in class. For the fourth task, modify the code for the fourth task to use a do-loop-until structure rather than a while loop. For all three tasks you should show the results of your function for cos(x) for x values of -1000, -314.159265358979, -1, 0, 1, and 31415.9265358979. (You can copy the input values of x with several digits from this assignment and paste them into the Excel workbook.) Use the Excel cosine function to verify that your answers are correct.

Accelerating convergence in tasks two to four Your code should recognize that cos(x) has a period of 2π, so that large user input values of x can be replaced by a smaller value of x, between zero and 2π, which will have the same value for the cosine. From the user input of a desired angle, x, you should be able to compute a value y whose absolute value is between zero and 2π. This smaller value, y, will have the same cosine as x and will allow the desired convergence to be obtained with fewer terms. This process, illustrated in the figure below, is described in the next paragraph.



Assume that we want to compute the cosine of 60 radians. (From the figure above we see that it would be close to -1, the same as the cosine of π radians.) The vertical dashed line between 50 and 60 radians in the figure passes through the cosine curve at a periodic repetition of its initial value of cos(0) = 1. This dashed line is located an integral multiple, N, of 2π from the origin. If we knew the value of N, we could subtract N times 2π from the value whose cosine we seek (60) and get a result, y, that would have the same cosine as x = 60. We can find the value of N by dividing x by 2π and taking the integer part; i.e., $N = \text{int}(x/(2\pi))$. In this example, $N = \text{int}(60/(2\pi)) = 9$ and $y = 60 - 2\pi N = 3.45133223538372$, which has the same cosine as x = 60, as you can verify on your calculator.

Since the cosine is a symmetric function, i.e. cos(-x) = cos(x), you can simply compute the cosine the absolute value of the input angle for negative angles.

You can take a further step to improve the efficiency of your calculations for angles greater than $\pi/2$ by using the following equations to compute a new value y between 0 and $|\pi/2|$ that will have the same value of cosine as the value of x for which you want to find the cosine:

$$\text{For } \pi/2 \le x \le 3\pi/2\text{: } \cos(x) = -\cos(y) \text{ where } y = (x - \pi)$$

$$\text{For } 3\pi/2 \le x \le 2\pi\text{: } \cos(x) = \cos(y) \text{ where } y = (2\pi - x)$$