

Make sure to write down your procedure clearly and answer exactly what the problem is asking for. Points will be taken off for incomplete, unintelligible, and sloppy procedures and solutions. Include comments in all MATLAB programs.

Follow the five step problem solving in engineering and science whenever required. Graphs are often useful ways to check your calculations.

**(20 points) 9.6** A Fibonacci sequence is composed of element created by adding the two previous elements. The simplest Fibonacci sequence starts with 1, 1 and proceeds as follows:

$$1, 1, 2, 3, 5, 8, 13, \dots$$

However, a Fibonacci sequence can be created with any two starting numbers. Fibonacci sequences appear regularly in nature. For example, the shell of the chambered nautilus (Figure 9.6) grows in accordance with a Fibonacci sequence. Prompt the user to enter the first two numbers in a Fibonacci sequence and the total number of elements requested for the sequence. Find the sequence and store it in an array by using a for loop. Now plot your results on a polar graph. Use the element number for the angle and the value of the element in the sequence for the radius.

**(20 points) 9.9** Recall from trigonometry that the tangent of both  $\pi/2$  and  $-\pi/2$  is infinity. This may be seen from the fact that

$$\tan(\theta) = \sin(\theta) / \cos(\theta)$$

and since

$$\sin\left(\frac{\pi}{2}\right) = 1$$

and

$$\cos\left(\frac{\pi}{2}\right) = 0$$

it follows that

$$\tan\left(\frac{\pi}{2}\right) = \text{infinity}$$

Because MATLAB uses a floating-point approximation of  $\pi$ , it calculates the tangent of  $\pi/2$  as a very large number, but not infinity.

Prompt the user to enter an angle  $\theta$  between  $\pi/2$  and  $-\pi/2$ , inclusive. If it is between  $\pi/2$  and  $-\pi/2$ , but not equal to either of those values, calculate  $\tan(\theta)$  and display the result in the command window. If it is equal to  $\pi/2$  or  $-\pi/2$ , set the result equal to Inf and display the result in the command window. If it is outside the specified range, send the user an error message in the command window and prompt the user to enter another value. Continue prompting the user for a new value of theta until he or she enters a valid number.

**(20 points) 9.13** Edmond Halley (the astronomer famous for discovering Halley's comet) invented a fast algorithm for computing the square root of a number, A.

Halley's algorithm approximates  $\sqrt{A}$  as follows:

Start with an initial guess  $x_1$ . The new approximation is then given by

$$Y_n = \frac{1}{A} x_n^2$$

$$x_{n+1} = \frac{x_n}{8} (15 - y_n(10 - 3y_n))$$

These two calculations are repeated until some convergence criterion,  $\epsilon$ , is met.

$$|x_{n+1} - x_n| \leq \epsilon$$

Write a MATLAB function called `my_sqrt` that approximates the square root of a number. It should have two inputs, the initial guess and the convergence criterion.

**(20 points) 9.15** The value of  $\sin(x)$  can be approximated as

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

Create a function called `my_sin`, using a midpoint break loop to approximate the value of  $\sin(x)$ . Determine convergence by comparing successive values of the summation as you add additional terms. These successive sums should be within an absolute value of 0.001 of each other. Test your function by evaluating the `my_sin(2)` and computing it to the built-in MATLAB sine function.

**(20 points) 9.20** Consider the following method to approximate the mathematical constant,  $e$ . Start by generating  $K$  uniform random integers between 1 and  $K$ . Compute  $J$ , the number of integers between 1 and  $K$ , which were never generated. We then approximate  $e$  by the ratio

$$K/J$$

Consider the following example for  $K=5$ . Assume that the following five integers are randomly generated between 1 and 5.

1 1 2 3 2

The number of times the integers are generated is given by

Integers	1	2	3	4	5
Number of instances	2	2	1	0	0

In this example, there are two integers, namely 4 and 5, which were never generated. This means that  $J=2$ . Consequently,  $e$  is approximated by

$$5/2=2.5$$

Write a function called `eapprox` that takes the value of  $K$  as input, and which then approximates  $e$  using the method described above. Test your function several times with different values of  $K$ , and compare the results to the value of  $e$  calculated using the built-in MATLAB function.

$$\exp(1)$$