Mathematics 2130
Project 2
Approximating π

The irrational constant π, used to represent the ratio between the circumference of a circle and its diameter, has a long and interesting history. By about 2000 BCE, the Babylonians and the Egyptians had discovered the constancy of this ratio. The Babylonians used $\frac{25}{8}$ as an approximation to it, while the Egyptians estimated it to be $4 \cdot \left(\frac{8}{9}\right)^2$. Subsequently, the more accurate value $\frac{22}{7}$ came to be used for π. Over the years, that value was refined to a considerable extent; nowadays, algorithms for computing (millions of) digits of π remain a standard benchmarking tool for testing new computer architectures. But interesting algorithms for computing π were devised long before the age of modern computing.

The purpose of this project is to explore the algorithm developed by the famous scientist and philosopher, Archimedes of Syracuse (circa 287–212 BCE). He described the oldest known algorithm which enables the calculation of π to any desired accuracy — a novel idea for his era!

Given a circle, Archimedes suggested that we **inscribe** a regular n-sided polygon — called an n-gon — inside the circle, and use the ratio of the perimeter of the polygon to the diameter as a lower bound for π. At the same time, he suggested that we also **circumscribe** a regular n-gon about the circle, and use the same calculation to obtain an upper bound for π. By choosing more sides (that is, making n larger) we can generate more accurate approximations to π.

Methodology

Archimedes' estimates of π were based on 96-gons. When you consider that he did this with no trigonometry, no systematic means of computing square roots, and without even a decimal number system, the feat becomes even more remarkable.

To begin, find a regular hexagon (a 6-gon) inscribed in the unit circle centred at the origin of a standard two-dimensional coordinate system. Calculate the length of each side of the hexagon and thereby compute a lower bound for π.

To proceed to larger values of n, imagine that the sides of the hexagon are rubber bands. To construct a 12-gon from a 6-gon, grab the midpoint of each of the hexagon's sides and pull those midpoints to the circle boundary in the centrifugal direction. We now have a dodecagon inscribed in the circle. What is the length of each side of this dodecagon? If you label a side of the hexagon ℓ_6 and a side of the dodecagon ℓ_{12}, then you can derive a formula for ℓ_{12} in terms of ℓ_6 using only the basic arithmetic operators $+$, $-$, $\times$, $\div$, $\sqrt{\ }$. In fact, all you need here is the Pythagorean theorem. After you do this, it should be clear how to proceed in an iterative fashion, so that you can determine the length of the sides of a regular 24-gon, then a 48-gon, and so on. Upper bounds to π can be obtained in an analogous way, using circumscribed polygons.

Archimedes started with a hexagon because, although he had no trigonometry, he knew that in a 30-60-90 triangle, the ratio of the short side to the hypotenuse is always $\frac{1}{2}$, that is, $\sin\left(\frac{\pi}{6}\right) = 0.5$. You may also make use of this fact. (Archimedes also required a means of computing square roots, but none was available. Like any good mathematician, he made one up!)

You may also start with an inscribed pentagon (a 5-gon) using the (less well-known) fact that

$$\cos\left(\frac{\pi}{5}\right) = \frac{1+\sqrt{5}}{4}.$$

In this approach, you will compute the perimeters of inscribed and circumscribed decagons (10-gons), 20-gons, ..., n-gons obtained by the iteration, as explained above. Or you could start with an inscribed square (a 4-gon) and utilise inscribed octagons (8-gons), 16-gons, and so on.

One of the tasks in this project is to write a computer program that will implement Archimedes' algorithm. Your program should include the following ingredients:

- The user should be asked for an initial value of n_0 (where n_0 is amongst a list of appropriate values). The program should then determine an estimate for π based on the perimeters of inscribed and circumscribed n_0-gons.

- Proceed to polygons with $n_1 = 2n_0$ sides, then polygons with $n_2 = 4n_0$ sides, and so on, up to a number of sides equal to $n_m = 2^m n_0$, where m is another user-inputted value. To reflect some of Archimedes' limitations, your program should only employ the basic arithmetic operators $+$, $-$, $\times$, $\div$, $\sqrt{\ }$. **Trigonometric functions must not be used in your program, nor in the development of the mathematics underlying it.**

Use your code to develop a list of upper and lower bounds for π based on sample values of n. Once you're sure that your program is working well, investigate how fast the difference between the upper and lower bounds decreases as n gets larger. Find out how fast these values approximate a high-precision value of π found elsewhere.

You can then use your code to verify other historical results about π. For instance, in the year 264 CE, a Chinese scholar named Liu Hui estimated $\pi = 3.14159$ using a 3072-gon. He independently developed the same algorithm as Archimedes, unaware that it had already been invented in Sicily 400 years earlier. Can you verify the accuracy of Lin Hui's calculation? Try to identify some other interesting episodes from the history of π (and not just those described on Wikipedia!).

Are there more general conclusions you can draw about the accuracy of Archimedes' algorithm? For instance, how many sides would be required to obtain ten-digit accuracy? What about d-digit accuracy for some integer d?

Note that suitable graphics are essential to this report. Your explanation of the mathematics underlying Archimedes' algorithm should make careful reference to clearly-drawn, suitably-labelled figures.