Exercise: Area Under the Curve

Borrowed from ACM Tech Pack 2 teaser (since I helped write it)

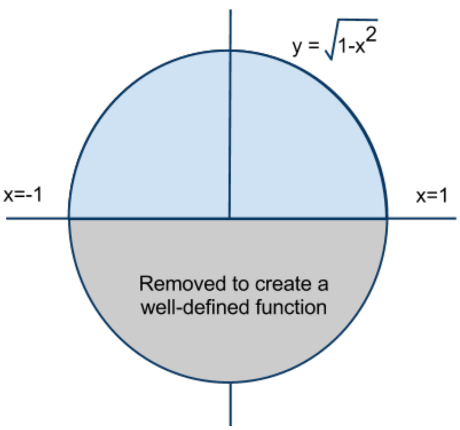
Numerical integration is an important technique for solving many different problems. To demonstrate the method, we utilize one type of numerical integration in order to calculate the value of Pi, since the end result is an easy one to compare to.
Implementations: serial, openmp, tbb, pthreads, winthreads, go, mpi, cuda, arbb

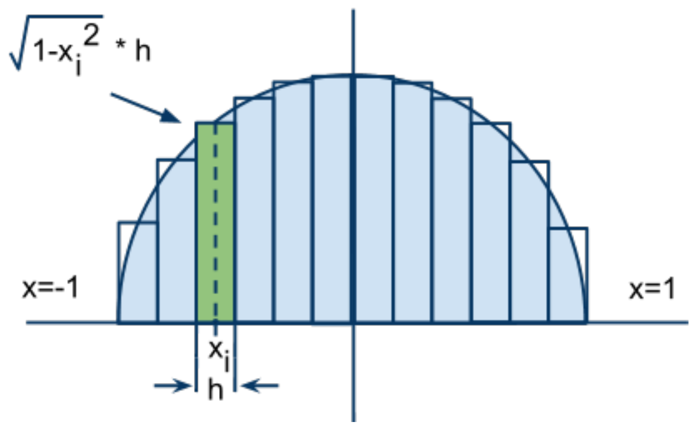
Problem definition

Often, we can estimate a desired quantity by finding the area under a curve (an integral). As an example of this type of computation, we will estimate the value π.

From grammar school, we know the area of a circle is πr^2.
Thus the area is just π for a unit circle.

From high school, we know the equation of a circle is $x^2 + y^2 = r^2$.
Thus the equation of a unit (semi)circle is $y = \sqrt{1-x^2}$





We will approximate this total area value (of the blue region in the diagram above) by adding up the areas of rectangles that approximately cover the area of that semicircle, as illustrated above.

This diagram shows 12 rectangles of equal width h that approximately cover the blue semicircular area. Each rectangle is positioned over a subinterval of the x-axis interval [-1,1], and the height of a rectangle is the function's value at some value xi in that rectangle's subinterval. Thus, the area of a rectangle is $\sqrt{1-x_i^2} * h$. We must add up the areas of all these rectangles, then double that sum to get the approximation of π. (What happens if we use the subinterval of the x-axis interval [0,1]?)

The more rectangles we use, the greater accuracy we expect in our rectangle approximation to the exact area under the semicircular curve y=f(x). Therefore, we will compute the sum with millions of thin rectangles in order to get better accuracy for our estimate of π. (Will things get more accurate if we use billions, trillions and why?)