OpenMP 1

1 Computation of π

Mathematically, we know that:

$$\pi = \int_0^1 \frac{4}{1+x^2}\, dx$$

Numerically, we can approximate the integral as the sum of rectangles (see Fig. 1):

$$\pi \approx \sum_{i=0}^{N} \frac{4}{1+x_i^2}\, \Delta x$$

where each rectangle has width Δx and height $F(x_i)$ at the middle of the interval i.

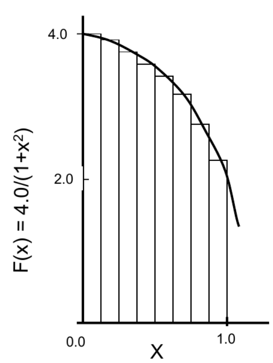

Figure 1: Numerical integration of π.

Task: Using only the OpenMP directives and functions seen in class so far:

- `#pragma omp parallel num_threads(...)`
- `omp_set_num_threads(...)`
- `omp_get_num_threads(...)`
- `omp_get_thread_num(...)`

Parallelize the code below so that 4 threads collaborate in the computation of π. You should manually divide the number of iterations among threads, and find a way for the local sums to survive the parallel region, to be accumulated afterwards by the master thread.

```
#include <stdio.h>
#include <stdlib.h>
#define NUM_STEPS 10000

int main( void )
{
  int i;
  double sum = 0.0, pi, x_i;
  double step = 1.0/NUM_STEPS;

  for ( i = 0; i < NUM_STEPS; i++ ) {
    x_i = (i + 0.5) * step;
    sum = sum + 4.0 / (1.0 + x_i * x_i);
  }
  pi = sum * step;
  printf("Pi: %.15e\n", pi);
  return 0;
}
```