**Kennesaw State University**

**Parallel and Distributed Computing**

**Project – MPI**

Instructor: Kun Suo

Points Possible: 100

Due date: check on the D2L

Mathematically, we know the following equation:

A close up of a logo

Description automatically generated

We can approximate the value of as a sum of rectangles:

Where each rectangle has width ∆x and height F(xi) at the middle of interval i.

The following code implements the above calculation of PI. We divide the area between 0 and 1 into 100000 small rectangles and the value of PI is approximately equal to the sum of all rectangles’ size. However, the program executes in the sequential implementation.

<https://github.com/kevinsuo/CS4504/blob/master/pi.c>

----------------------------------------------------------------------------------------------------------------

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define NUMSTEPS 1000000

int main() {

int i;

double x, pi, sum = **0.0**;

struct timespec start, end;

clock\_gettime(CLOCK\_MONOTONIC, &start);

double step = **1.0**/(double) NUMSTEPS;

x = **0.5** \* step;

**for** (i=**0**;i<= NUMSTEPS; i++){

x+=step;

sum += **4.0**/(**1.0**+x\*x);

}

pi = step \* sum;

clock\_gettime(CLOCK\_MONOTONIC, &end);

u\_int64\_t diff = **1000000000L** \* (end.tv\_sec - start.tv\_sec) + end.tv\_nsec - start.tv\_nsec;

printf("PI is %.20f**\n**",pi);

printf("elapsed time = %llu nanoseconds**\n**", (long long unsigned int) diff);

**return** **0**;

}

----------------------------------------------------------------------------------------------------------------

1. Write a parallel program to calculate PI using MPI based on this sequential solution.

To compile the program with MPI, use:

$ mpicc -g program.c -o program.o

Please write a brief report introducing your implementation.

1. Estimate your MPI program execution time and compare the time using different number of threads (n=1, 2, 4, …).

**Submitting Assignment**

Submit your assignment through D2L using the appropriate assignment link. For task, please submit the ***source code*** , ***screenshot of output*** and ***report****(introducing the code logic and highlight your results/output)*.