## 实验四 OpenMP编程

### 一、实验目的

（1）加深对并行计算及开发的理解。

（2）掌握基于OpenMP进行并行程序设计的方法。

### 二、实验内容

（1）掌握并行划分和计算的方法

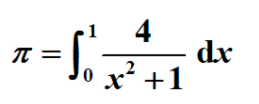
（2）使用OpenMP编写并执行用积分方法求π的并行程序。

### 三、实验原理及方案

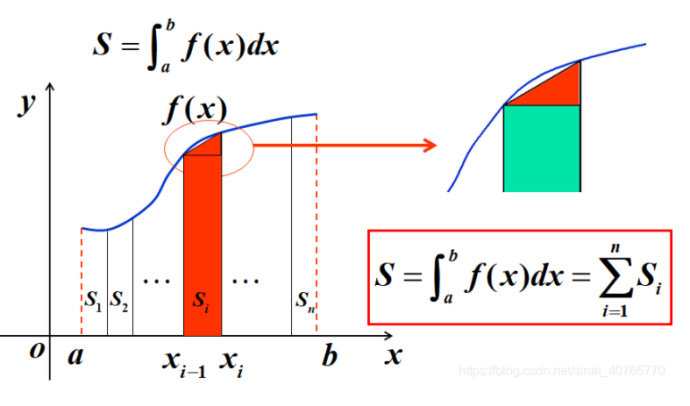
#### 1、实验环境设置

主流C/C++编译器（如gcc，g++等）大多支持OpenMP，可使用 #include <omp.h>调用OpenMP。

#### 2、π的求法

求π可用该积分方式求得：

应用积分的思想求解4/(x2+1)在0-1区间积分的值。将0-1积分区间分成n份，使n尽可能大。这时，被划分的每个小区域就可近似为一个小矩形，其宽为1/n，长可取小区域内的任意高度。求出每个小矩形的面积相加后即得到π值。



#### 3、并行划分与计算

选中各区间内中间位置的值作为小矩形的长。

则对于第k个小矩形，其中间位置为(a+b）/2 ：a、b为左右边界的位置，简单可知：a=k/n, b=(k+1)/n, 所以中间坐标j=(k+0.5)/n。则矩形的长为4/(j2+1)。

对于thread\_num个线程，简单分发划分的n个矩形计算任务。第k个任务分配到取余后的线程数上，即对于每个线程：Taskn+1 = Taskn + thread\_num。

最终结果为各线程结果相加。

#### 4、程序实现

设置线程数thread\_num、划分区间数n。设置线程数后，初始化线程域。

各个线程间不同的是执行任务的区别，所以在线程域初始化时，设置tid为各个线程私有，tid用omp\_get\_thread\_num()赋值为线程编号。由于所执行的任务有着Taskn+1 = Taskn + thread\_num的规律，可通过执行的第一个任务对其区别，第i个线程最开始都执行第i个任务。

根据跨度为thread\_num，循环计算，由于矩形宽固定为1/n，则只需将矩形长加和存储，最后再乘以1/n也得到矩形面积，计算量减小。线程域最后，各个线程分别输出其计算的值。

最后将各个线程长相加，并乘以step（1/n），即得到π的值。

### 四、实验要求

（1）做好实验预习。了解OpenMP的使用方法。

（2）根据实验原理及方案中的内容，完成实验。写出编程思路、源代码和实验结果。

（3）撰写实验报告。