|  |
| --- |
| #include <iostream> |
|  | #include <thread> |
|  | #include <chrono> |
|  | using namespace std; |
|  |  |
|  | double total\_sum = 0; |
|  | double total\_Thread = 0; |
|  | double f(double x) |
|  | { |
|  | return sin(x \* x) + x; |
|  | } |
|  |  |
|  | void simpson(double a, double b) { |
|  | auto start = std::chrono::system\_clock::now(); |
|  | double eps=0.01;//Нижний и верхний пределы интегрирования (a, b), погрешность (eps). |
|  |  |
|  | double I = eps + 1, I1 = 0;//I-предыдущее вычисленное значение интеграла, I1-новое, с большим N. |
|  | for (int N = 2; (N <= 4) || (fabs(I1 - I) > eps); N \*= 2) |
|  | { |
|  | double h, sum2 = 0, sum4 = 0, sum = 0; |
|  | h = (b - a) / (2 \* N);//Шаг интегрирования. |
|  | for (int i = 1; i <= 2 \* N - 1; i += 2) |
|  | { |
|  | sum4 += f(a + h \* i);//Значения с нечётными индексами, которые нужно умножить на 4. |
|  | sum2 += f(a + h \* (i + 1));//Значения с чётными индексами, которые нужно умножить на 2. |
|  | } |
|  | sum = f(a) + 4 \* sum4 + 2 \* sum2 - f(b);//Отнимаем значение f(b) так как ранее прибавили его дважды. |
|  | I = I1; |
|  | I1 = (h / 3) \* sum; |
|  | } |
|  | total\_sum += I1; |
|  | cout << I1 << endl; |
|  | auto end = std::chrono::system\_clock::now(); |
|  | std::chrono::duration<double> elapsed = end - start; |
|  | std::cout << "Elapsed time\_th: " << elapsed.count() << "s\n"; |
|  | total\_Thread += elapsed.count(); |
|  |  |
|  |  |
|  | } |
|  |  |
|  |  |
|  |  |
|  | int main() { |
|  | int a =0, b = 1000; |
|  | const int n = 10; |
|  | thread thrfirst[n]; |
|  | for (int i = 0; i < n; i++) { |
|  | thrfirst[i] = thread(simpson, a + b/n\*i, a + b/n \* (i+1)); |
|  | } |
|  | for (int i = 0; i < n; i++) { |
|  | thrfirst[i].join(); |
|  | } |
|  |  |
|  | cout << "SUM=" << total\_sum << endl; |
|  | double t\_th= total\_Thread; |
|  | cout << "Total sum of thread times: " << t\_th << endl; |
|  | total\_Thread = 0; |
|  | simpson(0, 1000); |
|  | std::cout << "\nOne Thread try:" << total\_Thread<<endl; |
|  | std::cout <<"\nWinned Time: " << t\_th/total\_Thread+ t\_th \* 100<<"%"; |
|  |  |
|  | return 0; |
|  | } |