

Лабораторная работа №1

Вариант 3

Выполнил: Эдуард Елисеев ИВТ-11М

Задача:

$$\int_{\frac{1}{2}}^1 \frac{6}{\sqrt{x(2-x)}} dx$$

Написать программу вычисления интеграла используя методы правых и левых треугольников.

Аналитическое решение:

$$12 \arcsin(\sqrt{x} / \sqrt{2}) + C$$

Определенный интеграл:

$$1 \text{ Pi}$$

Последовательная программа по расчету интеграла

```
C:\Users\epimetheus\source\repos\ips\lab1\x64\Debug\lab1.exe
Count of intervals = 100
Left integral = 3.1439227841
Accuracy (%) = 99.9258296424
Right integral = 3.1392817679
Accuracy (%) = 99.9264422253
Duration is: 0.0045827000 seconds

Count of intervals = 1000
Left integral = 3.1418248006
Accuracy (%) = 99.9926105304
Right integral = 3.1413606990
Accuracy (%) = 99.9926166563
Duration is: 0.0052673000 seconds

Count of intervals = 10000
Left integral = 3.1416158596
Accuracy (%) = 99.9992613287
Right integral = 3.1415694495
Accuracy (%) = 99.9992613900
Duration is: 0.0075840000 seconds

Count of intervals = 100000
Left integral = 3.1415949741
Accuracy (%) = 99.9999261356
Right integral = 3.1415903331
Accuracy (%) = 99.9999261362
Duration is: 0.0130443000 seconds

Count of intervals = 1000000
Left integral = 3.1415928856
Accuracy (%) = 99.9999926136
Right integral = 3.1415924215
Accuracy (%) = 99.9999926136
Duration is: 0.0857704000 seconds

Press any key to continue . . .
```

Программа по расчету интеграла с отключением векторизации

```
C:\Users\epimetheus\source\repos\ips\lab1\x64\Debug\lab1.exe
Count of intervals = 100
Left integral = 3.1439227841
Accuracy (%) = 99.9258296424
Right integral = 3.1392817679
Accuracy (%) = 99.9264422253
Duration is: 0.0058784000 seconds

Count of intervals = 1000
Left integral = 3.1418248006
Accuracy (%) = 99.9926105304
Right integral = 3.1413606990
Accuracy (%) = 99.9926166563
Duration is: 0.0040529000 seconds

Count of intervals = 10000
Left integral = 3.1416158596
Accuracy (%) = 99.9992613287
Right integral = 3.1415694495
Accuracy (%) = 99.9992613900
Duration is: 0.0105366000 seconds

Count of intervals = 100000
Left integral = 3.1415949741
Accuracy (%) = 99.9999261356
Right integral = 3.1415903331
Accuracy (%) = 99.9999261362
Duration is: 0.0131555000 seconds

Count of intervals = 1000000
Left integral = 3.1415928856
Accuracy (%) = 99.9999926136
Right integral = 3.1415924215
Accuracy (%) = 99.9999926136
Duration is: 0.0968415000 seconds

Press any key to continue . . .
```

Скорость выполнения слегка увеличилось, при том что точность результата не изменилась.

Программа по расчету интеграла с автоматическим параллелизатором

```
1>--- Analyzing function: double __cdecl left_rectangle_integral(double,double,int)
1>C:\Users\epimetheus\source\repos\ips\lab1\lab1\Source.cpp(29) : info C5001: loop vectorized
1>C:\Users\epimetheus\source\repos\ips\lab1\lab1\Source.cpp(34) : info C5002: loop not vectorized due to reason '1105'
1>C:\Users\epimetheus\source\repos\ips\lab1\lab1\Source.cpp(29) : info C5011: loop parallelized
1>C:\Users\epimetheus\source\repos\ips\lab1\lab1\Source.cpp(34) : info C5012: loop not parallelized due to reason '1000'
1>--- Analyzing function: double __cdecl right_rectangle_integral(double,double,int)
1>C:\Users\epimetheus\source\repos\ips\lab1\lab1\Source.cpp(51) : info C5001: loop vectorized
1>C:\Users\epimetheus\source\repos\ips\lab1\lab1\Source.cpp(56) : info C5002: loop not vectorized due to reason '1105'
1>C:\Users\epimetheus\source\repos\ips\lab1\lab1\Source.cpp(51) : info C5011: loop parallelized
1>C:\Users\epimetheus\source\repos\ips\lab1\lab1\Source.cpp(56) : info C5012: loop not parallelized due to reason '1000'
```

```
C:\> Select C:\Users\epimetheus\source\repos\ips\lab1\x64\Debug\lab1.exe

Count of intervals = 100
Left integral = 3.1439227841
Accuracy (%) = 99.9258296424
Right integral = 3.1392817679
Accuracy (%) = 99.9264422253
Duration is: 0.0061327000 seconds

Count of intervals = 1000
Left integral = 3.1418248006
Accuracy (%) = 99.9926105304
Right integral = 3.1413606990
Accuracy (%) = 99.9926166563
Duration is: 0.0055701000 seconds

Count of intervals = 10000
Left integral = 3.1416158596
Accuracy (%) = 99.9992613287
Right integral = 3.1415694495
Accuracy (%) = 99.9992613900
Duration is: 0.0057901000 seconds

Count of intervals = 100000
Left integral = 3.1415949741
Accuracy (%) = 99.9999261356
Right integral = 3.1415903331
Accuracy (%) = 99.9999261362
Duration is: 0.0137663000 seconds

Count of intervals = 1000000
Left integral = 3.1415928856
Accuracy (%) = 99.9999926136
Right integral = 3.1415924215
Accuracy (%) = 99.9999926136
Duration is: 0.0309185000 seconds
```

Со включенным флагом /Qpraг видно что скорость выполнения сократилась почти втрое при том что разница в точности результата настолько мала, что ей можно пренебречь

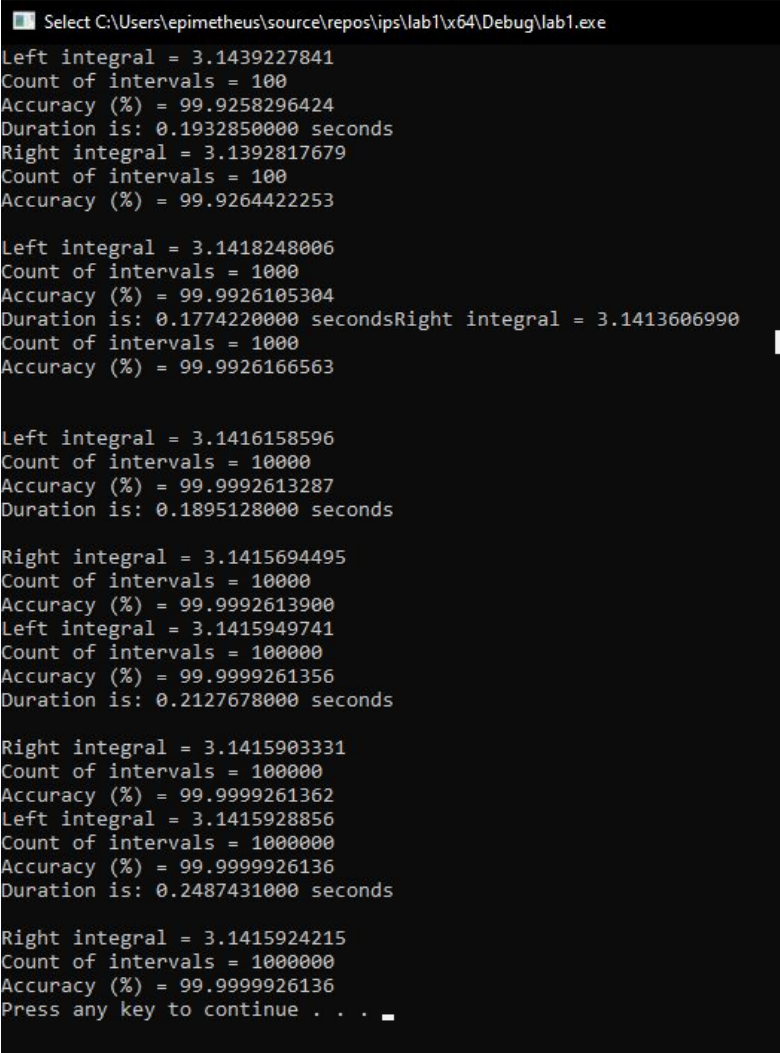
Итоговый набор флагов (прикладываю, так как простого подключения флага /Qpar и /Qpar-report не дало результатов, пришлось углубляться в значения других флагов Visual Studio):

```
/permissive- /ifcOutput "x64\Debug\" /GS /Qpar /W3 /Zc:wchar_t /ZI /Gm- /Ox /sdl  
/Fd"x64\Debug\vc142.pdb" /Zc:inline /fp:precise /D "_DEBUG" /D "_CONSOLE" /D  
"_UNICODE" /D "UNICODE" /errorReport:prompt /WX- /Zc:forScope /Gd /MDd /FC  
/Fa"x64\Debug\" /EHsc /nologo /Fo"x64\Debug\" /Fp"x64\Debug\lab1.pch"  
/diagnostics:column /Qpar-report:2 /Qvec-report:2
```

Тут, честно говоря, не совсем понял в чем суть задания. Если требуется сделать так, чтобы вычисление метода прямоугольников для каждого шага цикла проходило параллельно, то в таком случае я не придумал как передавать результат выполнения каждого вычисления обратно в вызывающего родителя, тут либо использовать

std::async - cppreference.com, либо создавать parallel_for, для которого тоже нужна доп. библиотека.

Сейчас я сделал проще, выведя просто всю функцию нахождения интеграла в отдельный поток вместе с используемым внутри циклом.



```
Select C:\Users\epimetheus\source\repos\ips\lab1\x64\Debug\lab1.exe  
Left integral = 3.1439227841  
Count of intervals = 100  
Accuracy (%) = 99.9258296424  
Duration is: 0.1932850000 seconds  
Right integral = 3.1392817679  
Count of intervals = 100  
Accuracy (%) = 99.9264422253  
  
Left integral = 3.1418248006  
Count of intervals = 1000  
Accuracy (%) = 99.9926105304  
Duration is: 0.1774220000 seconds  
Right integral = 3.1413606990  
Count of intervals = 1000  
Accuracy (%) = 99.9926166563  
  
Left integral = 3.1416158596  
Count of intervals = 10000  
Accuracy (%) = 99.9992613287  
Duration is: 0.1895128000 seconds  
  
Right integral = 3.1415694495  
Count of intervals = 10000  
Accuracy (%) = 99.9992613900  
Left integral = 3.1415949741  
Count of intervals = 100000  
Accuracy (%) = 99.9999261356  
Duration is: 0.2127678000 seconds  
  
Right integral = 3.1415903331  
Count of intervals = 100000  
Accuracy (%) = 99.9999261362  
Left integral = 3.1415928856  
Count of intervals = 1000000  
Accuracy (%) = 99.9999926136  
Duration is: 0.2487431000 seconds  
  
Right integral = 3.1415924215  
Count of intervals = 1000000  
Accuracy (%) = 99.9999926136  
Press any key to continue . . .
```

Здесь, как и следовало ожидать, время выполнения программы сократилось до минимума, так как просчет времени происходит в основном потоке, где не выполняется других инструкций. Также заметно что из-за того что вывод в cout происходит параллельно, иногда происходят коллизии по типу **Duration is: 0.1774220000 secondsRight integral = 3.1413606990.**

Ссылка на git: [ips_labs/lab1/lab1 at main · Epimetheus84/ips_labs \(github.com\)](https://github.com/Epimetheus84/ips_labs)