

## Лаба 3 по MPI Зверев И. Е.

Сделал все как было описано в пдфке

Работа с MPI происходит через MPIAdapter код в архиве

Что в архиве - решение на плюсах + немного кода на питоне для валидации

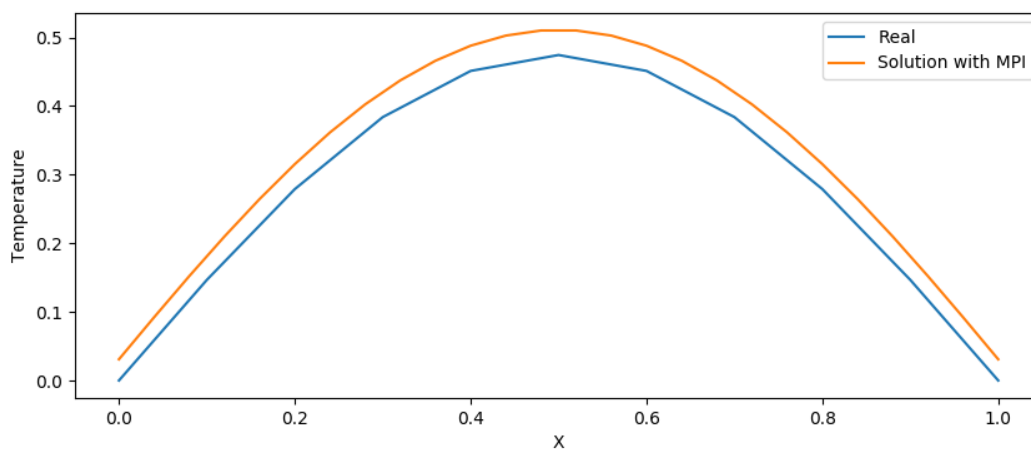
Ошибка для 11 точек такие:

Diff by points: [0.0308889, 0.03436946038847852, 0.036415632635625195,  
0.036080730210852785, 0.03675321271000298, 0.035727539620250826,  
0.03675321271000298, 0.036080730210852785, 0.03641563263562514,  
0.03436946038847846, 0.030888899999999994]

Mean square error: 0.001227696096649339

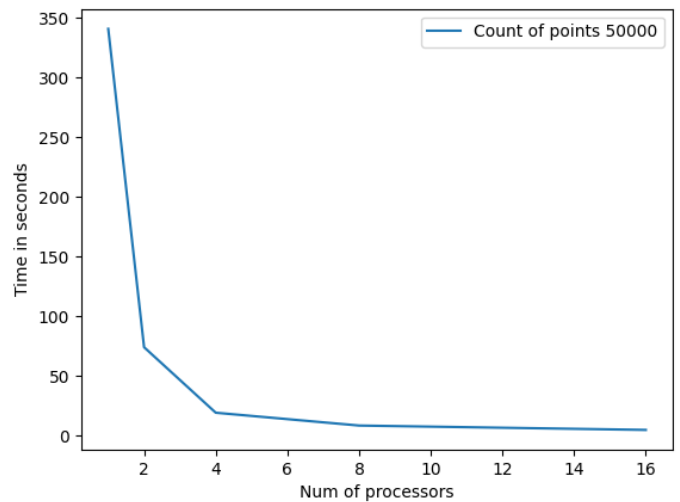
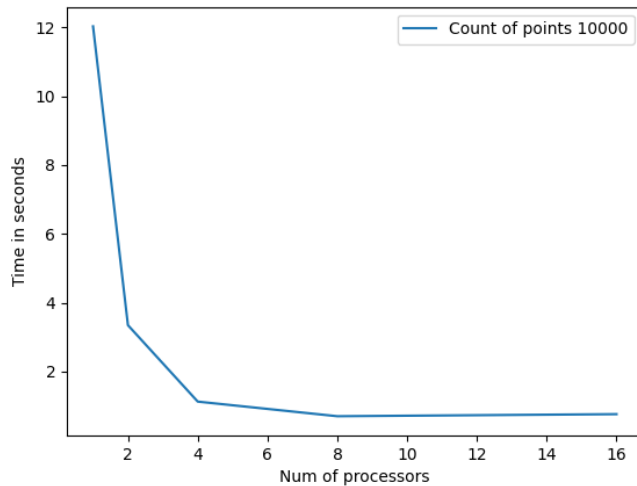
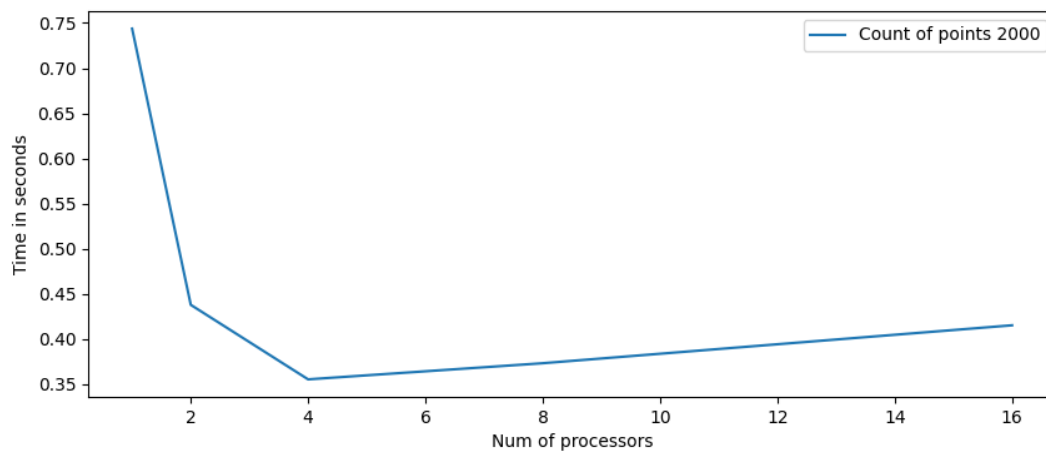
Abs error: 0.3847434115101696

Вот так выглядят ответы для решения с MPI и точного решения:



Чтобы запустить и получить картинку можно запустить `python3 make_graphics.py`  
Но оно будет считать сразу и графики для 2000, 10000 и 50000 точек  
После того как отработает получится 4 файла - `comparison.png` (картинка выше)  
`graph_2000.png`, `graph_10000.png` и `graph_50000.png` - графики времени работы для каждого N

Теперь графики времени работы для N = 2000, 10000, 50000



Как видим ускорение есть!

На машинке было 8 ядер, 8 потоков, можно заметить что на 16 процессах есть замедление

Само решение выглядит так - мастер процесс (с рангом 0) читает файл с заданием, далее делит их на задачи для разных процессов, потом рассылает их всем процессам и сам начинает считать свою часть, на каждом слое процессы обмениваются крайними значениями