Санкт-Петербургский политехнический университет Петра Великого

Институт прикладной математики и механики

**Кафедра «Теоретическая механика»**

**КУРСОВой проект**

**Решение двумерного уравнения теплопроводности при помощи MPI**

по дисциплине «Компьютерные технологии в механике»

Выполнил

студент гр. М.Г.Федоренко

Руководитель

Член-корреспондент РАН

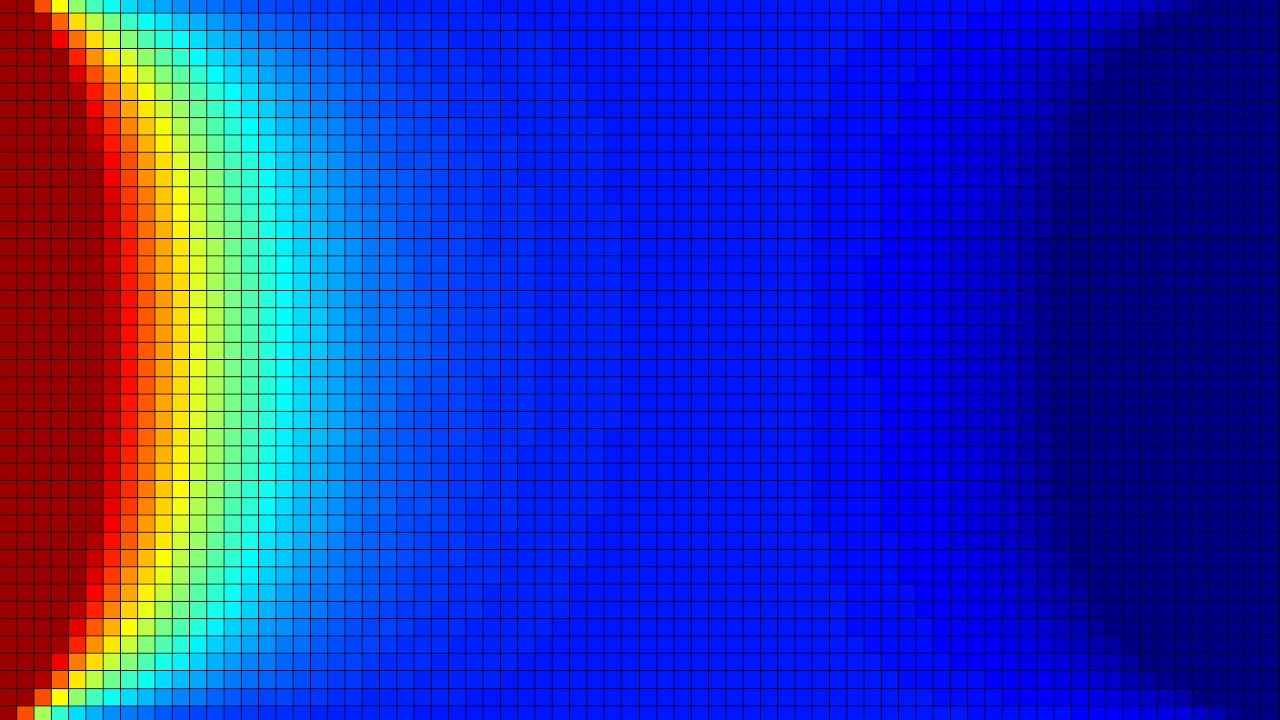
Заведующий кафедрой

"Теоретическая механика" А.М. Кривцов

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 201\_\_ г.

Санкт-Петербург

2017

****

**Постановка задачи:**

Необходимо решить задачу Коши для двумерного уравнения теплопроводности (дифференциальное уравнение в частных производных второго порядка, которое описывает распределение температуры в заданной области пространства и его изменение во времени.) с использованием средств параллельного программирования на основе MPI.

=

T(x,y,t)={T0 при x=0;

T1 при x! =0

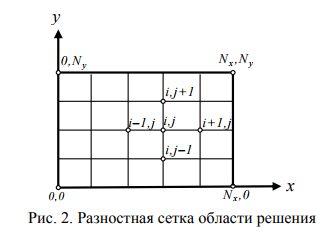
L=1 – размер объекта

= 200

= 0

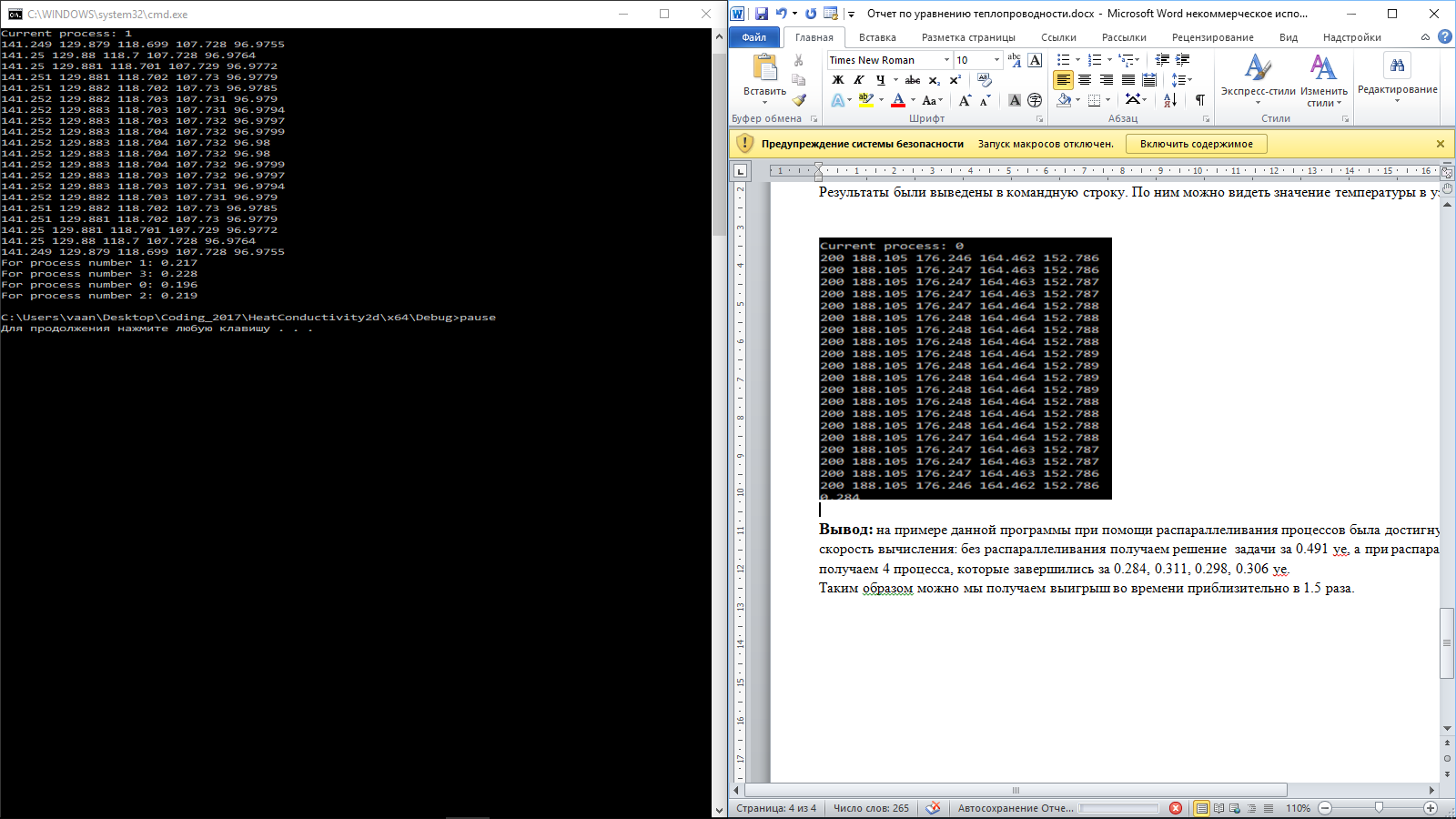
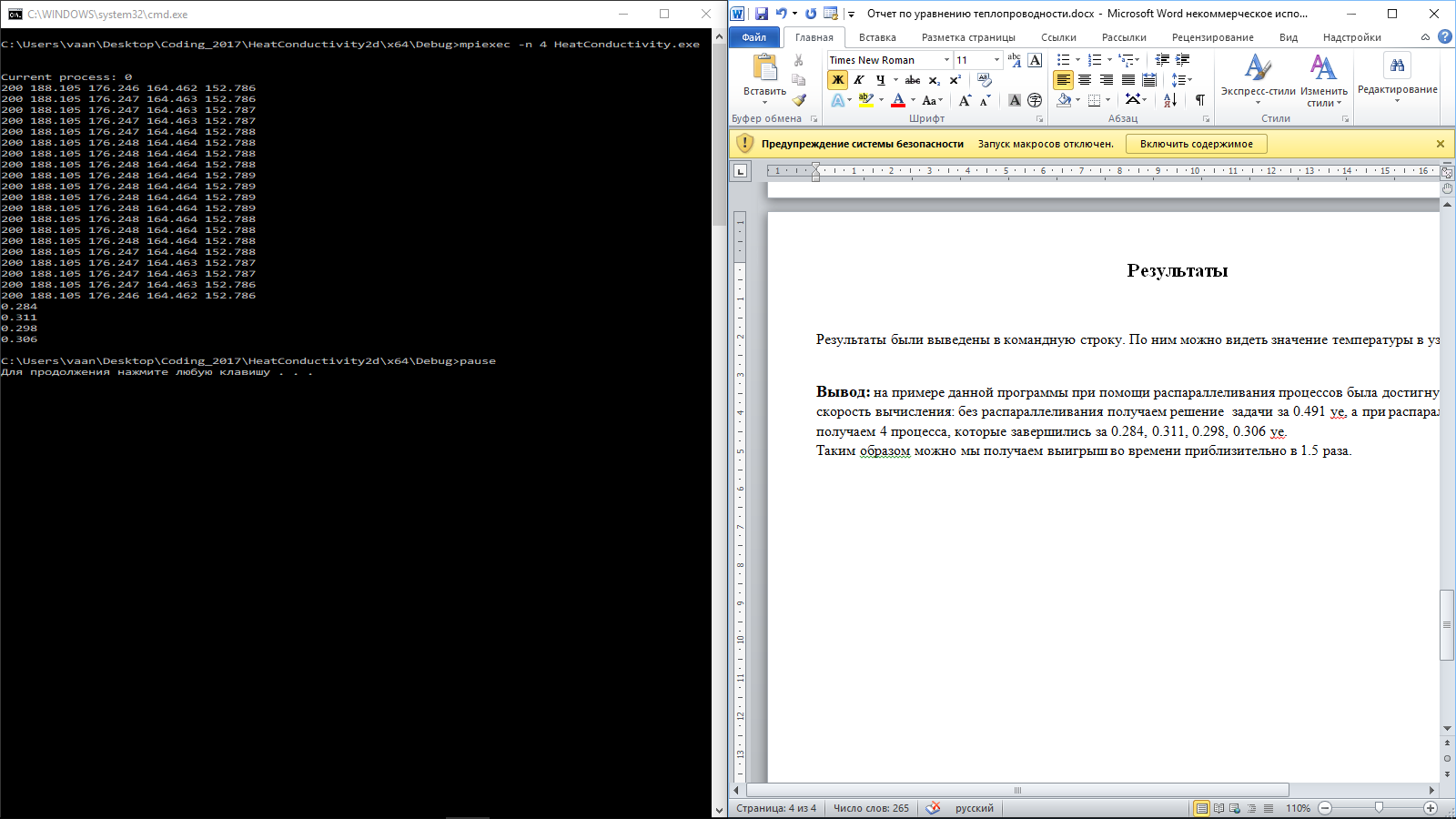
**Метод:**

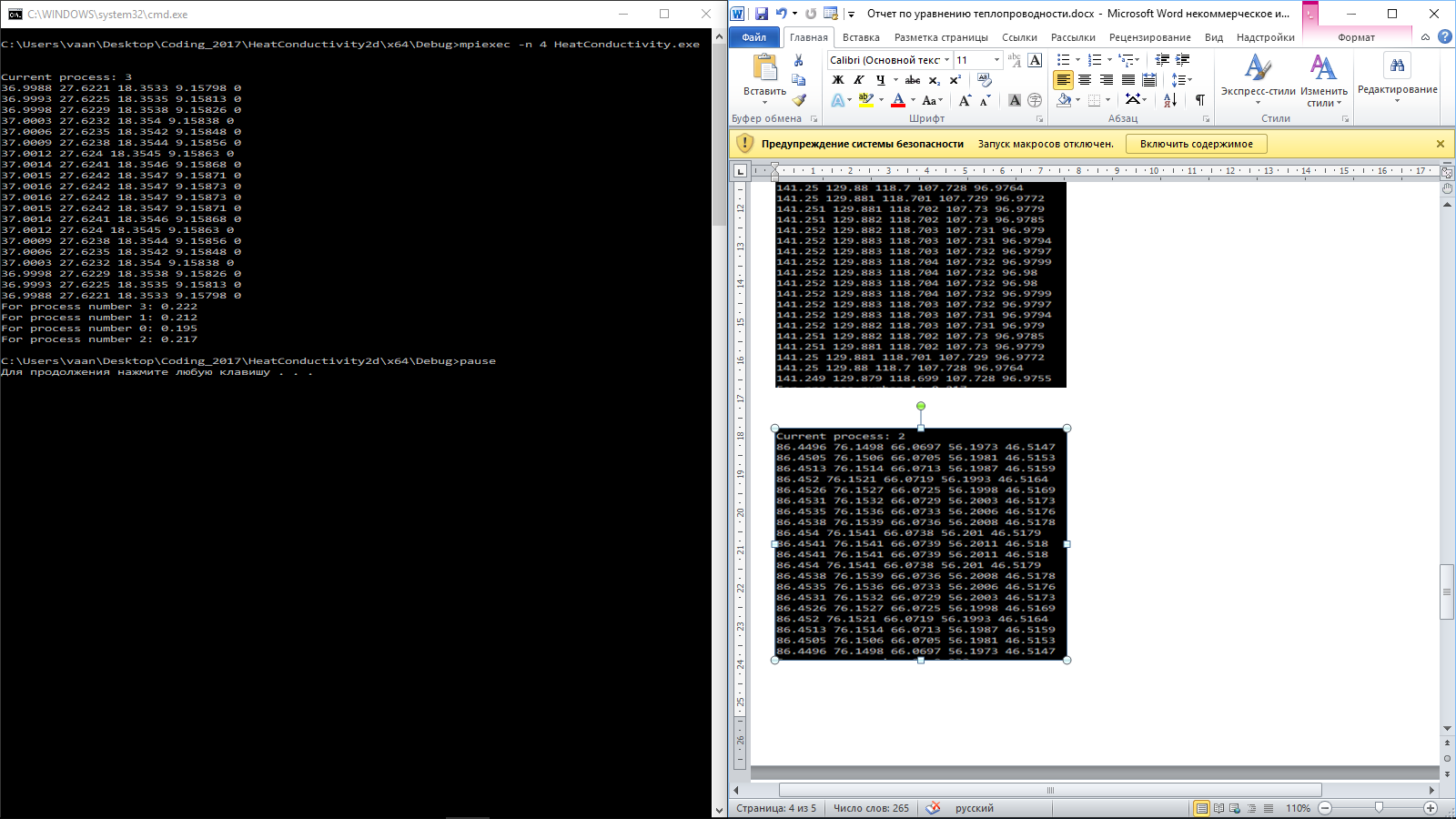
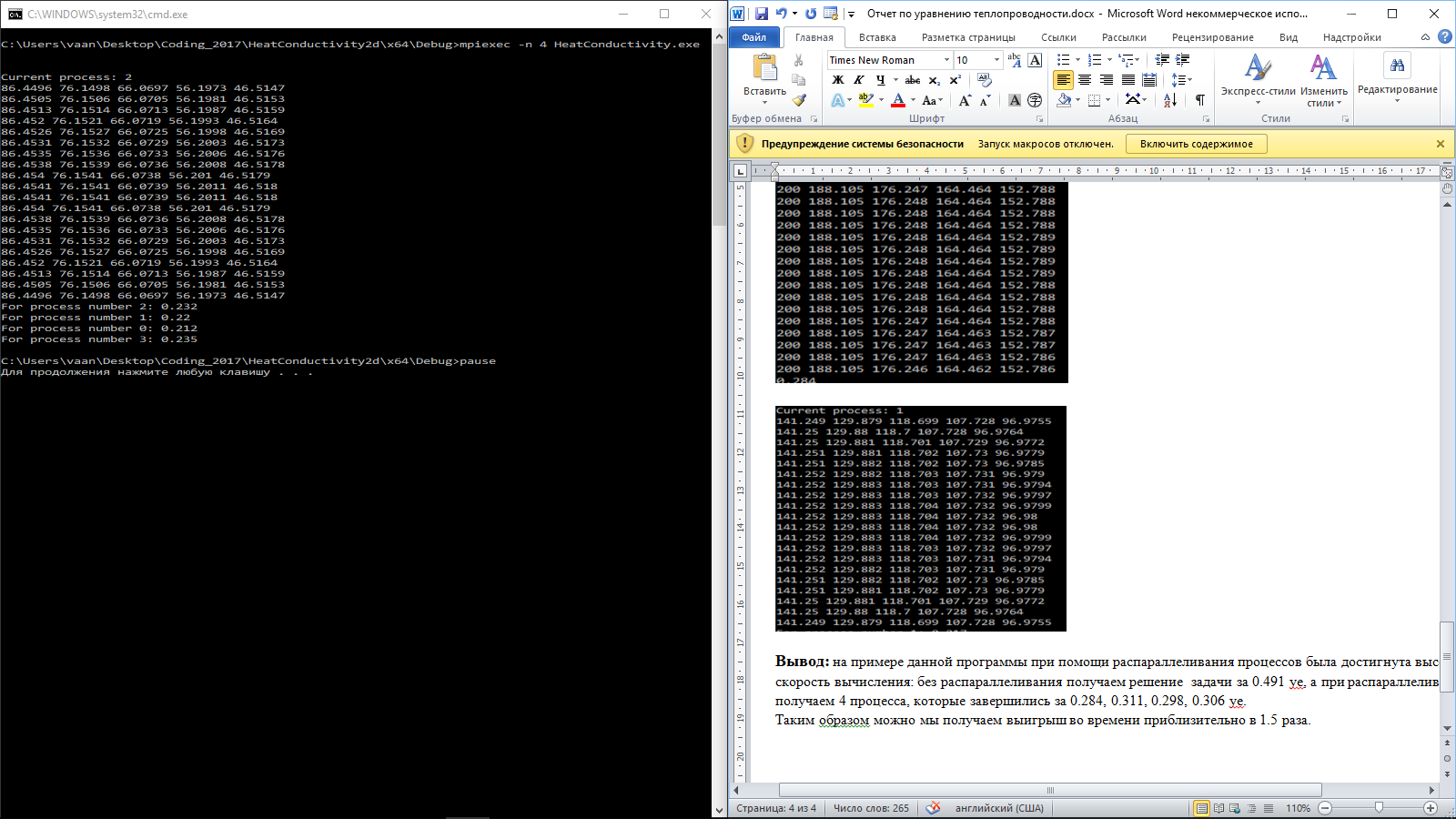
Параллельное программирование служит для создания программ, эффективно использующих вычислительные ресурсы за счет одновременного исполнения кода на нескольких вычислительных узлах. Для создания параллельных приложений используются параллельные языки программирования и специализированные системы поддержки параллельного программирования, такие как MPI и OpenMP. Итак, MPI - это библиотека передачи сообщений, собрание функций на C/C++ (или подпрограмм в Фортране), облегчающих коммуникацию (обмен данными и синхронизацию задач) между процессами параллельной программы с распределенной памятью. Под параллельной программой в рамках MPI понимается множество одновременно выполняемых процессов. Все процессы порождаются один раз, образуя параллельную часть программы. Каждый процесс работает в своем адресном пространстве, никаких общих переменных или данных в MPI нет. Процессы могут выполняться на разных процессорах, но на одном процессоре могут располагаться и несколько процессов (в этом случае их исполнение осуществляется в режиме разделения времени).



**Результаты**

Результаты были выведены в командную строку. По ним можно видеть значение температуры в узлах.





**Вывод**

На примере данной программы при помощи распараллеливания процессов была достигнута высокая скорость вычисления: без распараллеливания получаем решение задачи за 0.491 уе, а при распараллеливании получаем 4 процесса, которые завершились за 0.284, 0.311, 0.298, 0.306 уе.

Таким образом можно мы получаем выигрыш во времени приблизительно в 1.5 раза.