Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра интеллектуальных информационных технологий

**Отчет по лабораторной работе №2**

**по курсу «МРЗвИС»**

**на тему: «Реализация модели решения задачи на ОКМД архитектуре»**

|  |  |
| --- | --- |
| Выполнила студентка группы 021703: | Рабушка А.А. |
| Проверил: | Бруцкий Д.С. |

**МИНСК**

2022

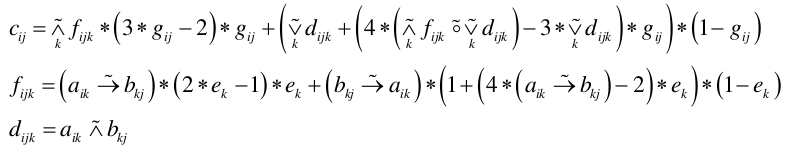
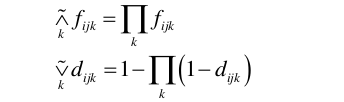
***Цель*:** реализовать и исследовать модель решения на ОКМД архитектуре задачи вычисления матрицы значений.

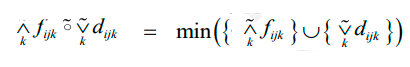
***Постановка задачи:***

*Дано*: сгенерированные матрицы ***A***, ***B***, ***E***, ***F***, ***G***, ***H*** заданных размерностей ***p***x***m***, ***m***x***q***, ***1***x***m***, ***p***x***q***, ***m***x***p*** и ***q***x***m*** соответственно со значениями в диапазоне [-1;1].

*Получить*: ***C*** – матрицу значений соответствующей размерности ***p***x***q***; в случае необходимости доопределить всеобщности(∀) или существования(∃) условие исходной задачи кванторами самостоятельно.

***Вариант 7:***





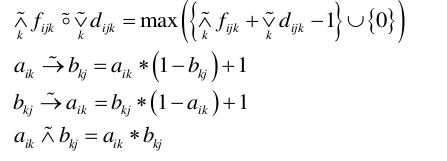
Описание модели

Была реализована модель решения на ОКМД архитектуре задачи вычисления матрицы значений. Возможность самостоятельно устанавливать все параметры, необходимые для работы модели, позволяет детально исследовать разработанную модель, установить зависимости между вышеуказанными параметрами. Также модель позволяет задавать условное время выполнения каждой из простейших операций (сложение, произведение, деление, сравнение, модуль), на основе заданного времени строятся графики, которые дают возможность наглядно увидеть, как изменяется коэффициент ускорения, эффективность и коэффициент расхождения программы от длительности операций.

Язык программирования, использованный для реализации модели: JavaScript.

# Исходные данные

Cгенерированные матрицы ***A***, ***B, E, G*** заполнены случайными числами в диапазоне [-1;1].

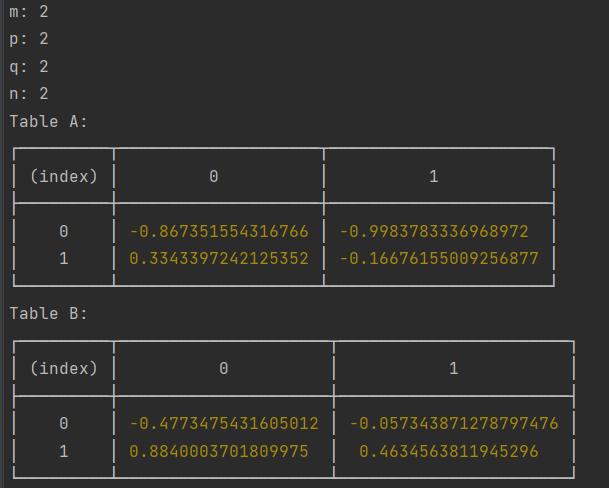
1. **p, m, q** – размерность матриц;
2. **n** – количество процессорных элементов в системе;
3. **ti** – время(длина) выполнения операции над элементами матриц.

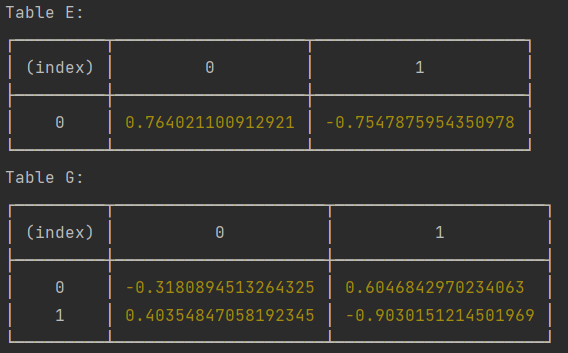
Для подсчета времени подсчитываются количества вызовов различных операций (сумма, разность и т.д.), а затем время одной операции умножается на количество вызовов данной операции, полученные значения суммируются. Для подсчёта необходимо установить зависимости между выполняемыми операциями. Если операции не являются зависимыми и их можно считать на различных процессорах, то время выполнение такой операции будет \*t , где 𝑡 – время выполнения такой операции, 𝑐 – количество вызовов данной операции, 𝑛 – количество процессоров, на которых выполняется операция. Если операции являются зависимыми, то время выполнение будет (𝑡1+𝑡2 +⋯ 𝑡𝑛), то есть сумма времен зависимых операций. При этом следует учесть, что результат некоторых зависимых операций можно получить, распараллелив её выполнение.

Для подсчета коэффициента расхождения задачи необходимо измерить две характеристики и, 𝐿𝑠𝑢𝑚 и 𝐿𝑎𝑣𝑔, где 𝐿𝑠𝑢𝑚- суммарная длина программы, a 𝐿𝑎𝑣𝑔 – средняя длина программы, то она будет равна 𝑇n. Чтобы посчитать 𝐿𝑎𝑣𝑔, необходимо знать, сколько объектов различных классов выполняется на каком-то из этапов вычислений. Данная задача была решена с помощью подсчета количества вызовов операций и функций на различных ветвях выполнения программой. Зная, количества объектов, выполняющихся на ветвях программы, время выполнения функции или операции, можно подсчитать 𝐿𝑎𝑣𝑔.

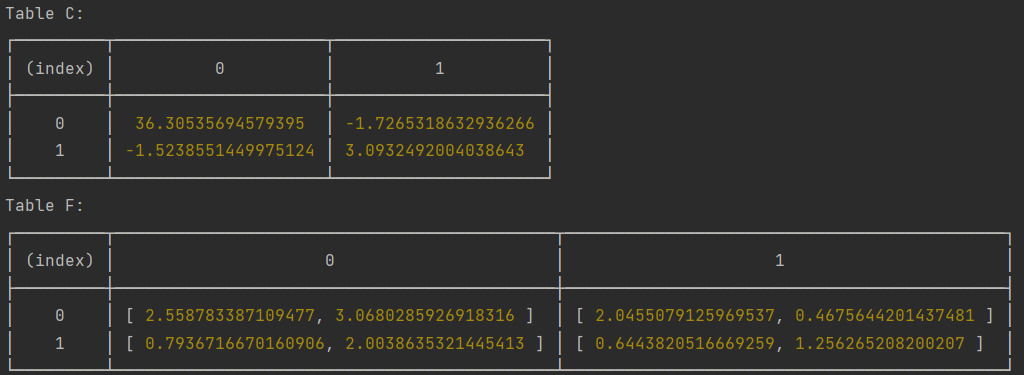
**Вопросы**:

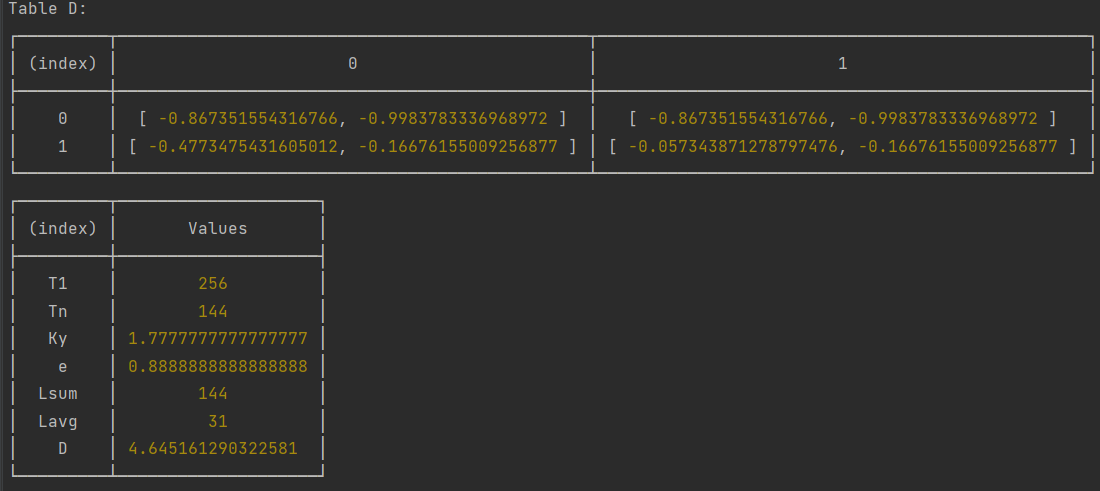
* 1. Проверить, что модель создана верно: программа работает правильно.





Результат



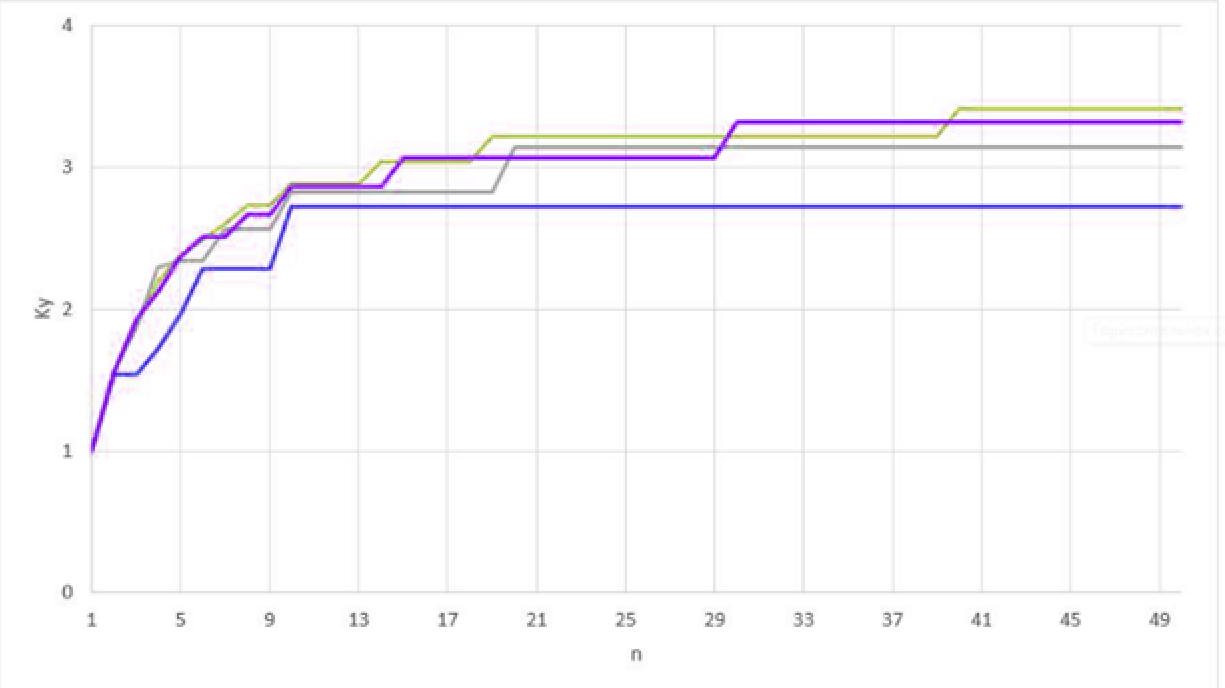


Вывод: модель создана верно

2. Построить графики и объяснить на них точки перегиба и асимптоты.

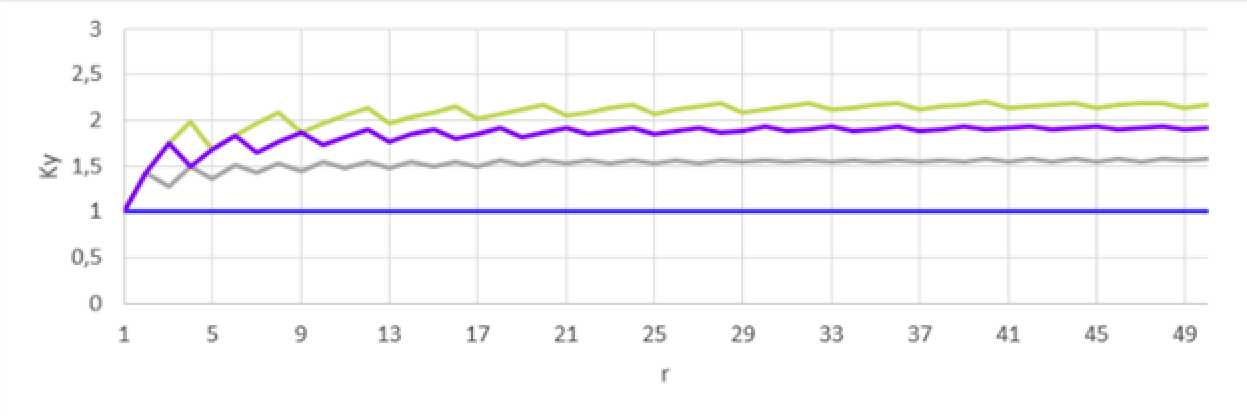
**Ку от n**:

Асимптотой графика является прямая,, которая параллельная оси абсцисс, ордината всех точек этой прямой равна значению ускорения при n = r. Как только количество процессорных элементов становится больше ранга задач, в вычислениях участвуют толь r процессорных элементов.

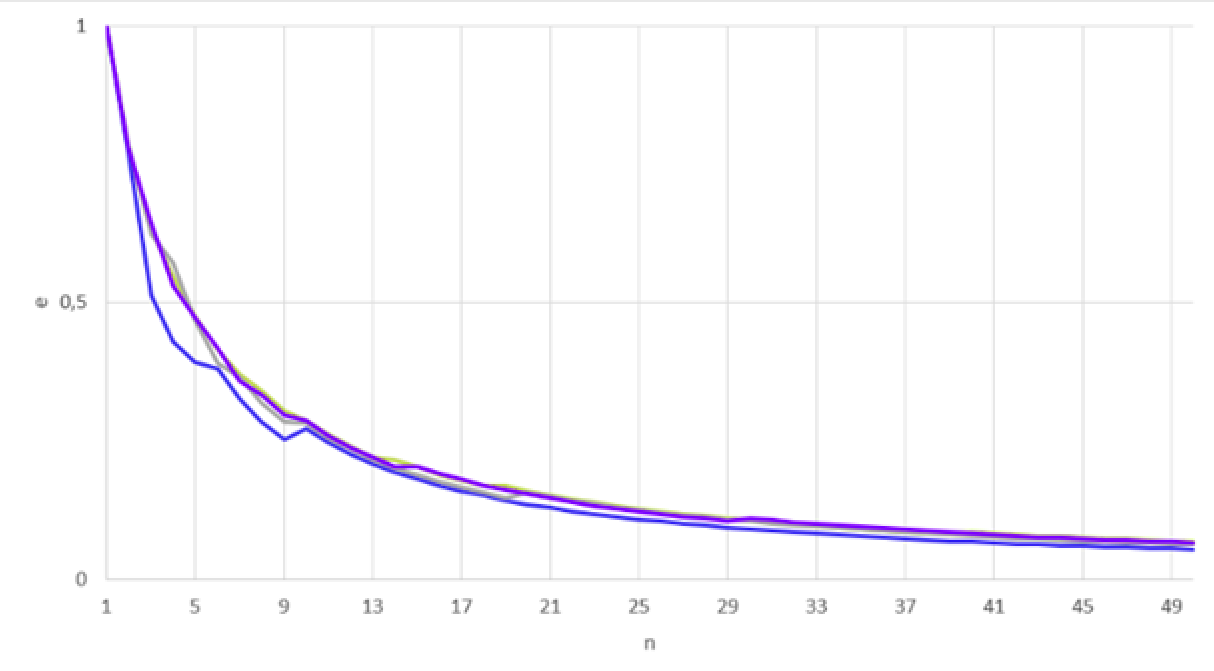


**Ку от r**:

Асимптотой является прямая , такого значения она достигает в точках , где ширина векторного параллелизма становится кратной числу процессорных элементов .

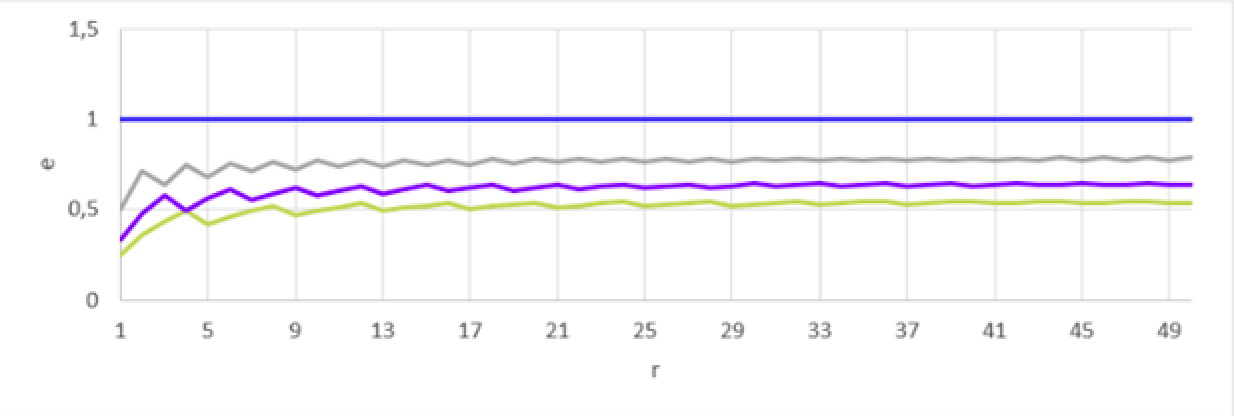


**e от n:**

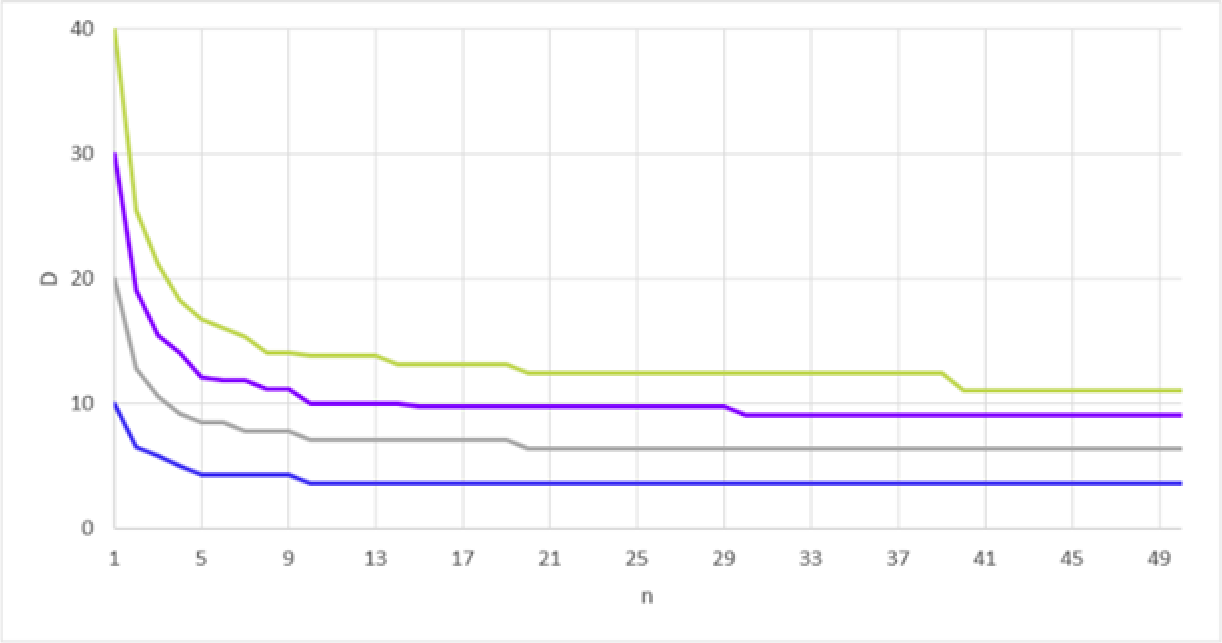
 Прямая будет являться асимптотой , это связано с тем , что в точке, где ширина векторного параллелизма становится равной числу процессорных элементов , перестает расти значение Ку

**e от r:**

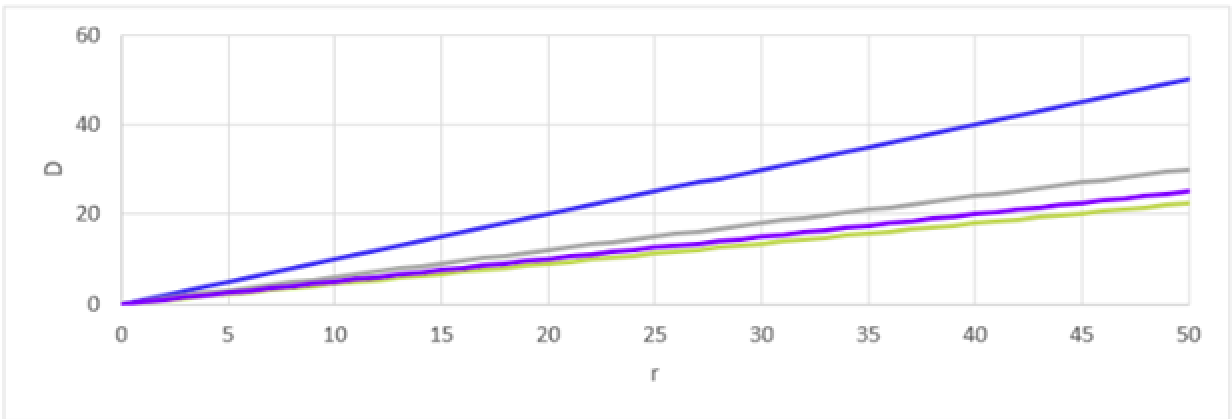
Прямая будет являться асимптотой, а точками перегиба – точки, где ширина векторного параллелизма становится кратной числу процессорных элементов



**D от n:**



**D от r:**



3.Спрогнозировать как изменится вид графиков при изменении параметров модели.

|  |  |
| --- | --- |
| **Семейства графиков** | **Как изменится вид графика** |
| График зависимости коэффициента ускорения Ку от n | При увеличении растёт коэффициент ускорения до точки, где ширина векторного параллелизма становится равной числу процессорных элементов, далее при увеличении коэффициент остаётся постоянным |
| График зависимости коэффициента ускорения Ку от r | При увеличении наблюдается скачкообразный рост коэффициента ускорения. Пиковые значения принимает в точках, где ширина векторного параллелизма становится равной числу процессорных элементов, в этих точках . |
| График зависимости эффективности e от n | При увеличении значения до точки , где ширина векторного параллелизма становится равной числу процессорных элементов, наблюдается скачкообразное изменение эффективности (падение-рост-падение). При дальнейшем увеличении эффективность падает. |
| График зависимости эффективности e от r | При увеличении наблюдается скачкообразное изменение эффективности (рост - падение - рост). Пиковые значения принимает в точках, где ширина векторного параллелизма становится кратной числу процессорных элементов |
|  | При увеличении значения до точки , где ширина векторного параллелизма становится равной числу процессорных элементов, наблюдается снижение коэффициент расхождения программы. |
|  | При увеличении наблюдается увеличение коэффициента расхождения программы. |

# Выводы.

Была реализована модель решения на ОКМД архитектуре задачи вычисления матрицы значений. Реализованная модель была проверена на работоспособность и правильность получаемых результатов. Данная модель позволяет ускорить процесс вычисления матрицы значений.

Были исследованы числовые характеристики ОКМД архитектуры, а именно коэффициент ускорения, эффективность и коэффициент расхождения программы при решении поставленной задачи.