

Московский государственный университет имени М.В.Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра математических методов прогнозирования

**Теоретическая работа №1 по курсу  
«Суперкомпьютерное моделирование и  
технологии»**

**Построение информационного графа и  
определение свойств заданного фрагмента.**

**Выполнил:**

студент 617 группы

Г.В. Кормаков

Москва, 2021

# Содержание

<b>1</b>	<b>Постановка задачи</b>	<b>2</b>
<b>2</b>	<b>Информационная структура</b>	<b>2</b>
<b>3</b>	<b>Анализ графового представления</b>	<b>4</b>
<b>4</b>	<b>Сведения об информационной структуре</b>	<b>6</b>
<b>5</b>	<b>Разметка параллельных циклов</b>	<b>10</b>
<b>6</b>	<b>Выводы</b>	<b>11</b>

# 1 Постановка задачи

Для выполнения задания был выдан представленный фрагмент программы (Листинг 1):

```
1  for(i = 2; i <= n+1; ++i)
2      C[i] = C[i - 1] + D[i];
3
4  for(i = 2; i <= n+1; ++i)
5      for(j = 2; j <= m+1; ++j)
6          B[i][j] = B[i + 1][j - 1];
7
8  for(i = 2; i <= n+1; ++i){
9      A[i][1][1] = B[i][m + 1] + C[n + 1];
10     for(j = 2; j <= m+1; ++j)
11         for(k = 1; k <= n; ++k)
12             A[i][j][k] = A[i][j - 1][1] + A[i][j][k];
13 }
```

Листинг 1: program\_106.c

Необходимо было выполнить исследование информационной структуры этого фрагмента, рассмотреть связи по операциям, выполняемым в данном фрагменте и определить характеристики, важные для понимания степени параллелизма фрагмента.

Информационную структуру фрагмента составлена на языке разметки Algolang (см. [4]). Для визуализации структуры использовался инструмент Algoview(см. [3]).

## 2 Информационная структура

Важными для формирования информационной структуры фрагмента оказались следующие факты:

1. Узлами формируемой структуры являются операторы (чаще всего - это операторы присваивания). Зависимость по данным рассматривает последовательность обращений в память, однако для оценки информационной структуры более информативным является зависимость по времени исполнения.
2. При отсутствии зависимостей от предыдущих данных структура фрагмента будет совпадать с описанной ниже, однако, если бы перед фрагментом программы присутствовали вычисления, связанные с элементами  $D[i]$ , то необходимо было бы отразить зависимости в структуре.

3. Циклические зависимости (например, при присвоении  $A[i][j] = A[i][j] + D[i]$ ) не отслеживаются, т.к. по времени нет зависимости между одной и той же областью памяти в явном виде.

Описание информационной структуры приведено ниже на языке Algolang (см. листинг 2). Для моделирования в среде Algoview параметры были подобраны так, чтобы была возможность рассмотреть структуру.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <algo>
3   <params>
4     <param name="n" type="int" value="5"> </param>
5     <param name="m" type="int" value="4"> </param>
6   </params>
7
8   <!-- C[i] = C[i - 1] + D[i]-->
9   <block id="0" dims="1">
10     <arg name="i" val="2..n+1"> </arg>
11     <vertex condition="" type="1">
12       <in src="i - 1"> </in>
13     </vertex>
14   </block>
15
16   <!-- B[i][j] = B[i + 1][j - 1]-->
17   <block id="1" dims="2">
18     <arg name="i" val="2..n+1"> </arg>
19     <arg name="j" val="2..m+1"> </arg>
20     <vertex condition="" type="1">
21       <in src="i - 1, j + 1"> </in>
22     </vertex>
23   </block>
24
25   <!-- A[i][1][1] = B[i][m + 1] + C[n + 1]-->
26   <block id="2" dims="1">
27     <arg name="i" val="2..n+1"> </arg>
28     <vertex condition="" type="1">
29       <in bsrc="1" src="i, m + 1"> </in>
30       <in bsrc="0" src="n + 1"> </in>
31     </vertex>
32   </block>
33
34   <!-- A[i][j][k] = A[i][j - 1][1] + A[i][j][k]-->
35   <block id="3" dims="3">
36     <arg name="i" val="2..n+1"> </arg>
37     <arg name="j" val="2..m+1"> </arg>
38     <arg name="k" val="1..n"> </arg>
39     <vertex condition="(j == 2)" type="1">
40       <in bsrc="2" src="i"> </in>
41     </vertex>
42     <vertex condition="(j != 2)" type="1">
43       <in src="i, j - 1, 1"> </in>
44     </vertex>
45   </block>
46 </algo>

```

Листинг 2: algoload\_106.xml

### 3 Анализ графового представления

Представленный в листинге 2 фрагмент был загружен в систему AlgoLoad (см. [5]) с собственного профиля. Были получены результаты, скриншоты которых приведены ниже (рис. 1, 2, 3).

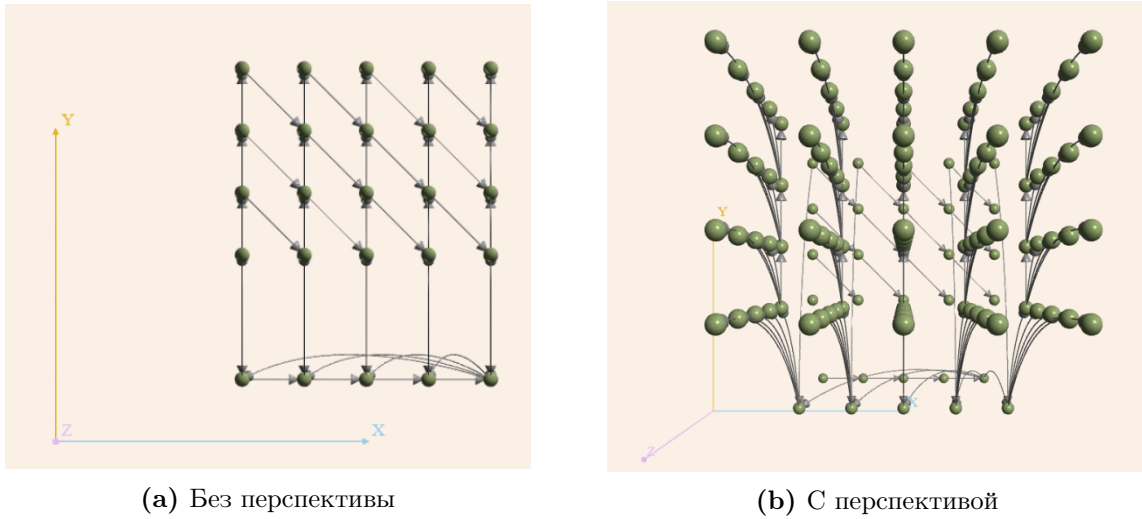


Рис. 1: Проекция на плоскость XY

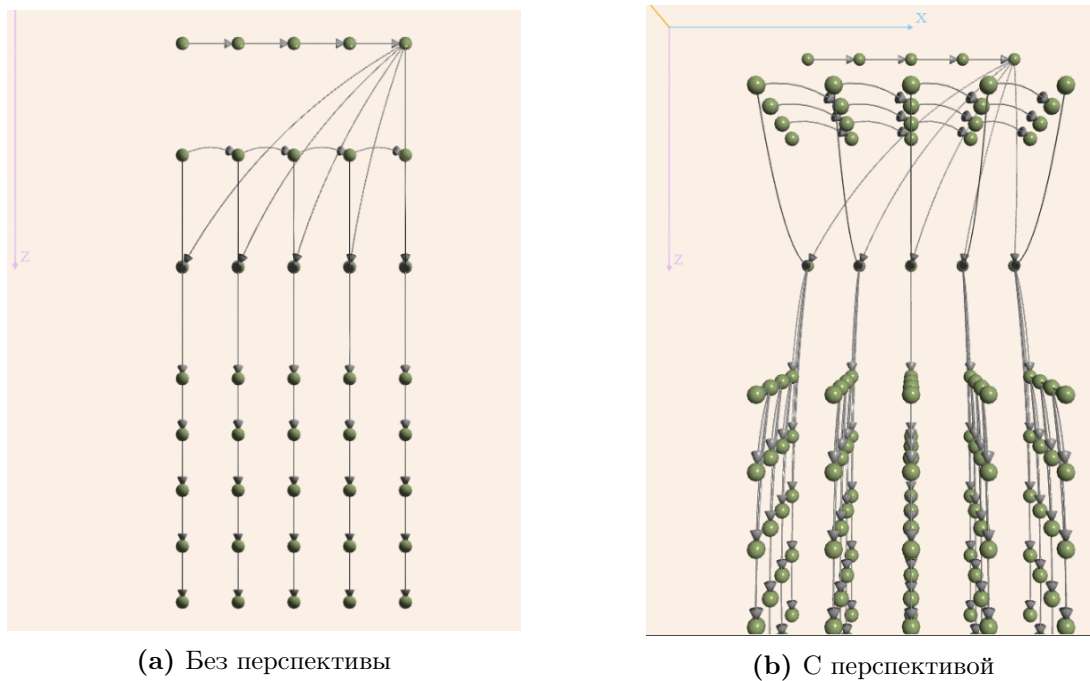
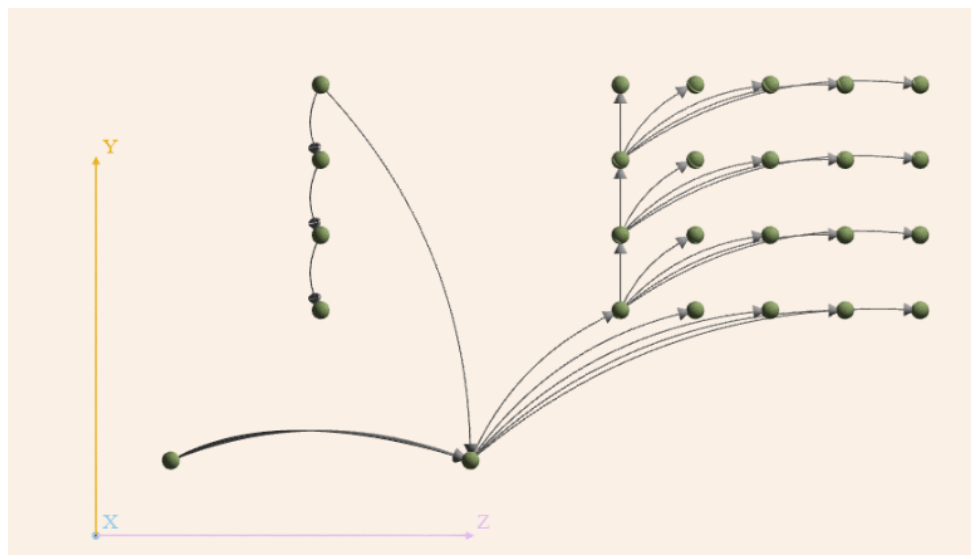
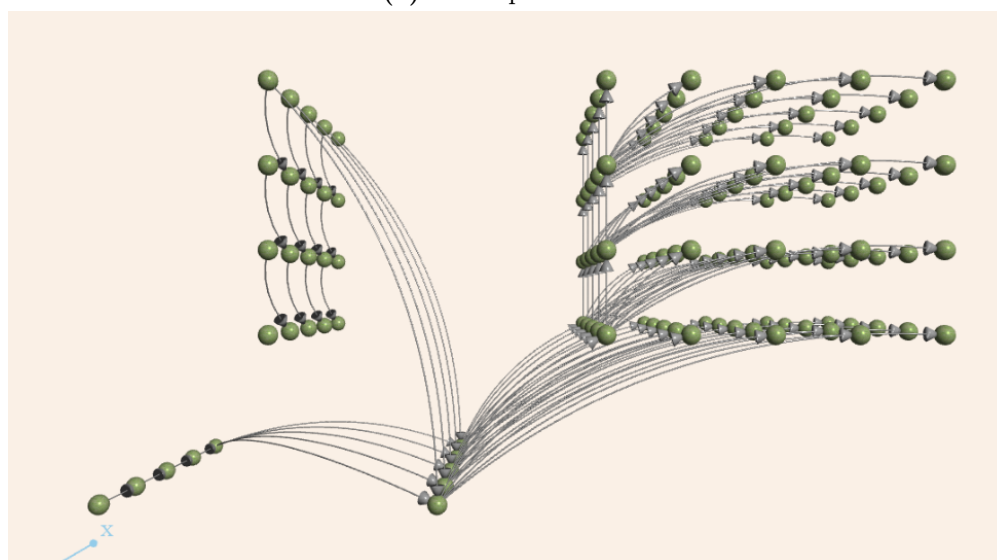


Рис. 2: Проекция на плоскость XZ



(a) Без перспективы



(b) С перспективой

**Рис. 3:** Проекция на плоскость YZ

Наиболее информативной с визуальной точки зрения является проекция на плоскость YZ (рис. 3). Совместим информативное описание фрагмента с рисунком для понимания ресурсов параллелизма и соответствия с фрагментами (рис. 4).

Для уточнения зависимостей в цикле размерности 3 на рис. 4 используем условие на вторую размерность, как и в приведённом фрагменте листинга 2. На изображении общими группами также отмечены условным образом пути, связанные с зависимостями в циклах. Некоторые из них соответствуют ресурсам параллелизма (основа для формирования выводов - [2]).

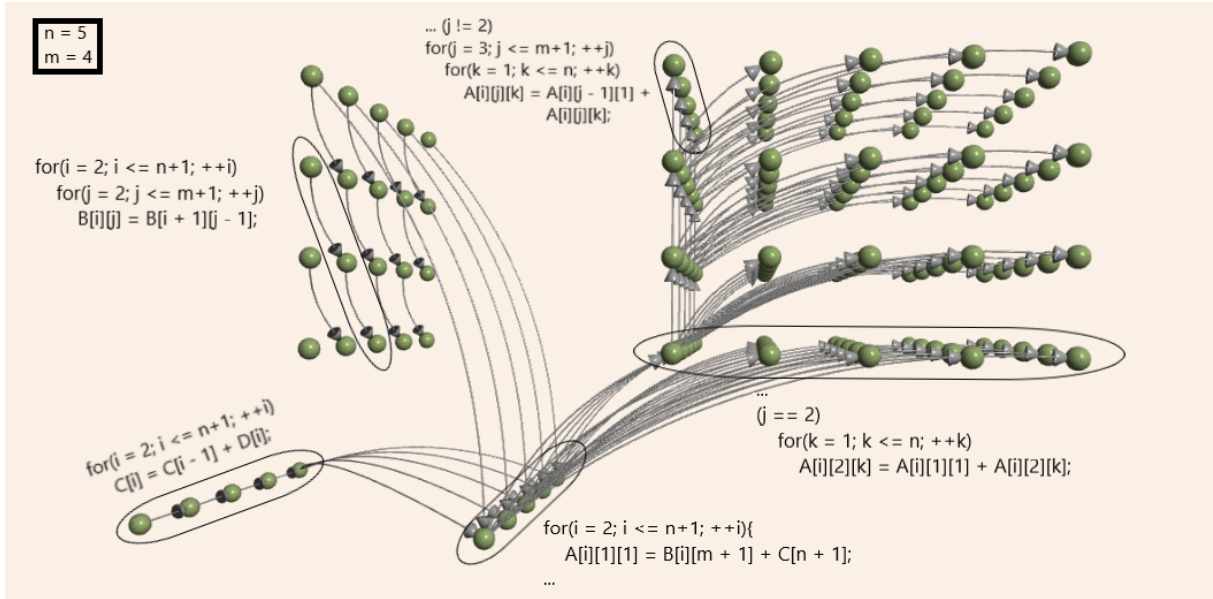


Рис. 4: Комментарии к структуре информационного графа

## 4 Сведения об информационной структуре

Выход алгоритма Algoview:

Total vertex count: 130, Critical route length: 21, Canonical LPF width: 25

1. **Последовательная сложность** (число вершин в информационном графе фрагмента). Результата работы программы совпадает с собственными подсчётами. Число вершин = 130. В общем случае формула имеет следующий вид:  

$$n + nm + n + nmn = 2n + nm + n^2m = \mathbf{n(nm + m + 2)} = 5 \cdot (20 + 4 + 2) = 130.$$
2. **Параллельная сложность** (длина (число дуг) критического пути в информационном графе). Результат собственных вычислений отличается от результата программы. Результат программы - 21 (не смог найти объяснений, почему результат Algoview такой). Собственный подсчёт - 9. Общая формула собственного подсчёта:  $(n - 1) + 1 + (n - 1) = \mathbf{2n - 1} = 9.$
3. **Ширина ЯПФ** (максимальное число вершин на ярусе). Результат алгоритма совпадает с собственными вычислениями. Ширина ЯПФ = 25. **Данная параллельная форма является канонической** (все входные вершины находятся на ярусе 1 и на каждом ярусе k длина максимального пути до любой вершины этого яруса равна k-1). И, следовательно, максимальной (т.е. минимальной высоты).

На рис. 5 приведён один из ярусов с максимальной шириной. Общая схема подсчёта числа вершин на этом ярусе равна:  $n^2 = 25$ .

Благодаря ещё одному свойству, критический путь является минимальной из всех возможных высот ЯПФ - 1 (поскольку высота ЯПФ - число вершин). Таким образом, можно подтвердить, что критический путь равен 9 для данных параметров (факт взят из знаний предыдущего курса и может быть найден в [1]).

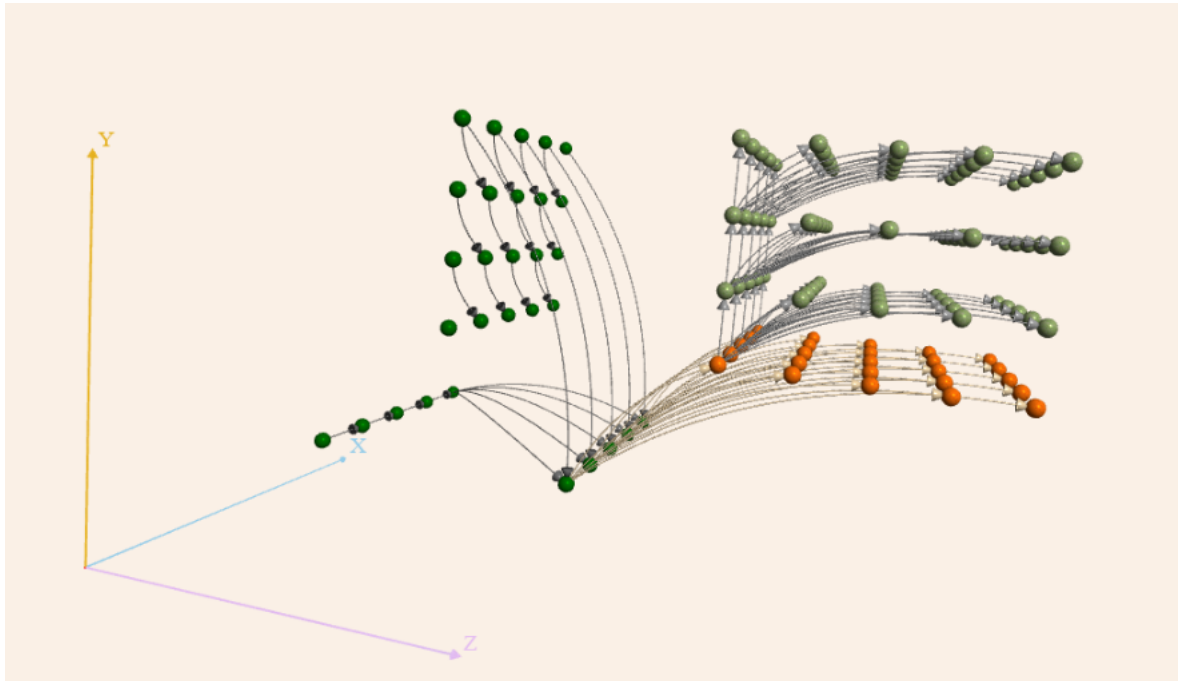


Рис. 5: Вид яруса с максимальной шириной (выделен оранжевым)

4. **Максимальная глубина вложенности циклов** = 3.
5. **Число различных типов дуг** (тип дуг определяется направляющим вектором и длиной при фиксированных значениях параметров).

Все дуги, имеющие одинаковые направляющие векторы и одинаковую длину, имеют один тип. Если направляющие векторы и длины зависят от внешних параметров, то мы фиксируем параметры.

Стоит отметить, что ранее не рассматривались данные понятия, поэтому интерпретировать данные определения (направляющего вектора и длины) будем по связи с теорией графов и эвристиками...



Направляющие векторы разделим (типизация частично опирается на классификацию in и out из [1]) на типы по:

- Принадлежности входа и выхода дуги одному блоку итераций
- Принадлежности входа и выхода дуги одной размерности блока итераций

В этом случае по второму критерию происходит разбиение дуг на типы по входам/выходам в различные размерности. Таким образом, может получиться, что число типов дуг зависит от размерности.

Для приведённого фрагмента типы дуг отчётливо видны на рис. 5. **Всего типов:**  $1 + 5 + 1 + 1 + 5 + 5 = 1 + n + 1 + 1 + n + n = 3(n + 1) = 18$ .

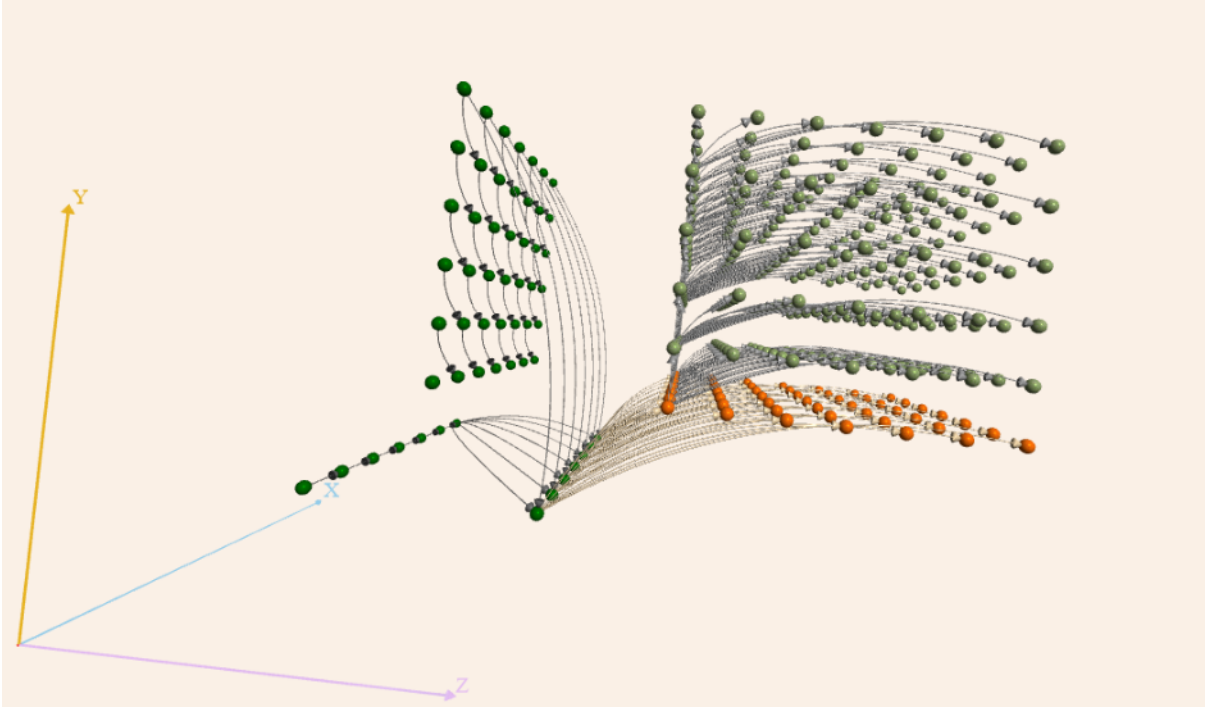
- 1 тип - по размерности  $X$  в блоке  $C[i] = C[i - 1] + D[i]$
- $n$  типов - по выполнению операции  $A[i][1][1] = B[i][m + 1] + C[n + 1]$  от последнего элемента предыдущего блока
- 1 тип - по операции  $B[i][j] = B[i + 1][j - 1]$  (диагональные направляющие векторы по размерностям)
- 1 тип - по операции  $A[i][1][1] = B[i][m + 1] + C[n + 1]$  (от последнего элемента блока итераций массива  $B$ )
- $n$  типов - операции  $A[i][2][k] = A[i][1][1] + A[i][2][k]$
- $n$  типов - операции  $A[i][j][k] = A[i][j - 1][1] + A[i][j][k]$  ( $j \neq 2$ )

6. **Наличие длинных дуг** (т.е. дуг, длины которых зависят от внешних параметров). Внешними параметрами заданного фрагмента являются  $n$  и  $m$  (не считая зависимость по матрице  $D$ ).

Соответственно, зависимыми от размерности являются дуги операций:

- $A[i][1][1] = B[i][m + 1] + C[n + 1]$  (от последнего элемента блока итераций массива  $B$ ) ( $n = 5$  дуг, т.к. хоть они и одного типа, но общее количество их равно  $n$ )
- $A[i][1][1] = B[i][m + 1] + C[n + 1]$  от последнего элемента предыдущего блока ( $n = 5$  дуг)
- $A[i][2][k] = A[i][1][1] + A[i][2][k]$  ( $n^2 = 25$  дуг)
- $A[i][j][k] = A[i][j - 1][1] + A[i][j][k]$  ( $j \neq 2$ ) ( $n^2(m - 1) = 100$  дуг)

ОДНАКО, если подумать над характером зависимости, то именно длины изменяются у дуг типа 6a. Остальные же (типов 6b - 6d) просто появляются при увеличении размерности. Да, они длинные в визуальном плане, однако зависимость длины от размера параметра фиксирована для каждой из них. Этот факт визуально продемонстрирован на рис. 6 благодаря увеличению параметров  $n$  и  $m$  (можно сравнить с рисунком 5).



**Рис. 6:** Увеличение длин дуг при увеличении размерностей  $n - > 7, m - > 6$

Таким образом, более правильным ответом на вопрос о количестве длинных дуг считаю  $\mathbf{n} = 5$ .<sup>1</sup> Если считать зависимости вида  $length(dim) = const(dim), \forall dim \in \mathbf{N}$ , где  $const(dim)$  – константы, однозначно определённые размерностью, входящими в определение «зависимости длины от размерности», то верный ответ  $= n + n + n^2 + n^2(m - 1) = 2n + n^2m$ .

---

<sup>1</sup>Этот факт опирается только на приведённые рассуждения. В приложенном к заданию примере отчёта не содержится достаточно информации для однозначного определения количества длинных дуг. Т.к. в отчёте пример программы не содержит возникающих дуг при увеличении размерности с таким же типом, как в предложенном для анализа здесь.

## 5 Разметка параллельных циклов

Поскольку условия задания не предусматривают изменения данного фрагмента, то разметка на параллельные циклы проводится только по ресурсу параллелизма, доступному в данном фрагменте. Полезными источниками для понимания теоретических возможностей (без визуального представления) распараллеливания циклов являются [6],[7].

Для обоснования разметки фрагмента на параллельные циклы воспользуемся определениями **вектора расстояний**  $\mathbf{D}$  и **вектора направлений**  $\mathbf{d}$  для многомерных циклов (см. [7]).

Для цикла размерности 1 (с массивами C и D):

- $\mathbf{D} = 1$  (расстояние между стоком (sink) и источником (source))
- Между операторами цикла *потокосная зависимость* и (т.к.  $\mathbf{D} = 1$ ) цикл *не может быть распараллелен*

Для цикла размерности 2 (с массивом B):

- $\mathbf{D} = (-1, 1)$
- $\mathbf{d} = (" > ", " < ")$  (полное совпадение с примером 6.6 [7])
- Между операторами *антизависимость*. Распараллеливание по внутреннему циклу возможно всегда. Для распараллеливания по двум одновременно или по внешнему циклу необходимо резервирование данных, что требует изменение структуры фрагмента, следовательно, не размечается.

Для цикла размерности 3 (с массивами A,B,C):

- $\mathbf{D} = (0, > 0, 0)$  <sup>2</sup>
- $\mathbf{d} = (" = ", " < ", " = ")$

---

<sup>2</sup>В явном виде расписать значение второй компоненты вектора расстояний затруднительно для цикла в данном виде. Однако, поскольку по  $i$  есть возможность распараллеливания, то можно утверждать о положительности, т.к. значения по второй размерности используются, как минимум, на следующей итерации по этой размерности.

- Между операторами существует *истинная зависимость* по второй размерности. Распараллеливание по 1 и 3 размерности возможно всегда. Распараллеливание по 2 размерности невозможно.

Размеченный на основании этих выводов фрагмент приведён в листинге 3.

```

1  for(i = 2; i <= n+1; ++i)
2      C[i] = C[i - 1] + D[i];
3
4  for(i = 2; i <= n+1; ++i)
5      #pragma omp parallel for
6          for(j = 2; j <= m+1; ++j)
7              B[i][j] = B[i + 1][j - 1];
8
9  #pragma omp parallel for
10 for(i = 2; i <= n+1; ++i){
11     A[i][1][1] = B[i][m + 1] + C[n + 1];
12     for(j = 2; j <= m+1; ++j)
13         #pragma omp parallel for
14             for(k = 1; k <= n; ++k)
15                 A[i][j][k] = A[i][j - 1][1] + A[i][j][k];
16 }
```

Листинг 3: pragma\_program\_106.c

## 6 Выводы

(Почти все дальнейшие рассуждения выходят за рамки обязательного задания. Поэтому для ускорения проверки можно их пропустить, но хотелось бы узнать, если будет возможность, на сколько рассуждения верны)

1. Оценим возможную степень параллелизма данного фрагмента. Доля последовательных операций равна  $\alpha = \frac{n+nm+nm+1}{n(nm+m+2)} = \frac{46}{130} \approx 0.354$  (для заданных параметров)<sup>3</sup>. Для оценки, примем  $\alpha \approx \frac{1}{n}$  (это удовлетворяет порядку предела)

Тогда, по закону Амдала,  $S \leq \frac{1}{\alpha + \frac{1-\alpha}{p}} \approx n$

2. Данный фрагмент также имеет смещённый ресурс параллельности в блоке размерности 2. На практике он может быть оптимизирован.
3. Понятия типа дуги и зависимости длины от размерности должны быть формализовано на этапе постановки задачи, иначе может получиться несоответствие результатов.

---

<sup>3</sup>При увеличении n и m доля стремится к 0?

## Источники

- [1] В. В. Воеводин, Вл. В. Воеводин. Параллельные вычисления Спб, изд-во «БХВ-Петербург», 2002.
- [2] Математические основы параллельных вычислений.  
[https://academy.hpc-russia.ru/files/math\\_foundations.pdf](https://academy.hpc-russia.ru/files/math_foundations.pdf)
- [3] Описание системы Algoview.  
[https://parallel.ru/sites/default/files/info/education/instrukciya\\_po\\_rabote\\_s\\_algoview.pdf](https://parallel.ru/sites/default/files/info/education/instrukciya_po_rabote_s_algoview.pdf)
- [4] Описание языка Algolang.  
[https://parallel.ru/sites/default/files/info/education/opisanie\\_yazyka\\_algolang.pdf](https://parallel.ru/sites/default/files/info/education/opisanie_yazyka_algolang.pdf)
- [5] Описание системы Algoload.  
[https://parallel.ru/sites/default/files/info/education/instrukciya\\_po\\_rabote\\_s\\_algoload.pdf](https://parallel.ru/sites/default/files/info/education/instrukciya_po_rabote_s_algoload.pdf)
- [6] Лобанов А.И. Лекция 5. Зависимости в простых циклах и их анализ на параллельность.  
[http://hpc-education.ru/files/lectures/2011/lobanov/lobanov\\_2011\\_lecture05.pdf](http://hpc-education.ru/files/lectures/2011/lobanov/lobanov_2011_lecture05.pdf)
- [7] Лобанов А.И. Лекция 6. Зависимости во вложенных циклах и их анализ на параллельность.  
[http://hpc-education.ru/files/lectures/2011/lobanov/lobanov\\_2011\\_lecture06.pdf](http://hpc-education.ru/files/lectures/2011/lobanov/lobanov_2011_lecture06.pdf)