Параллельные вычисления

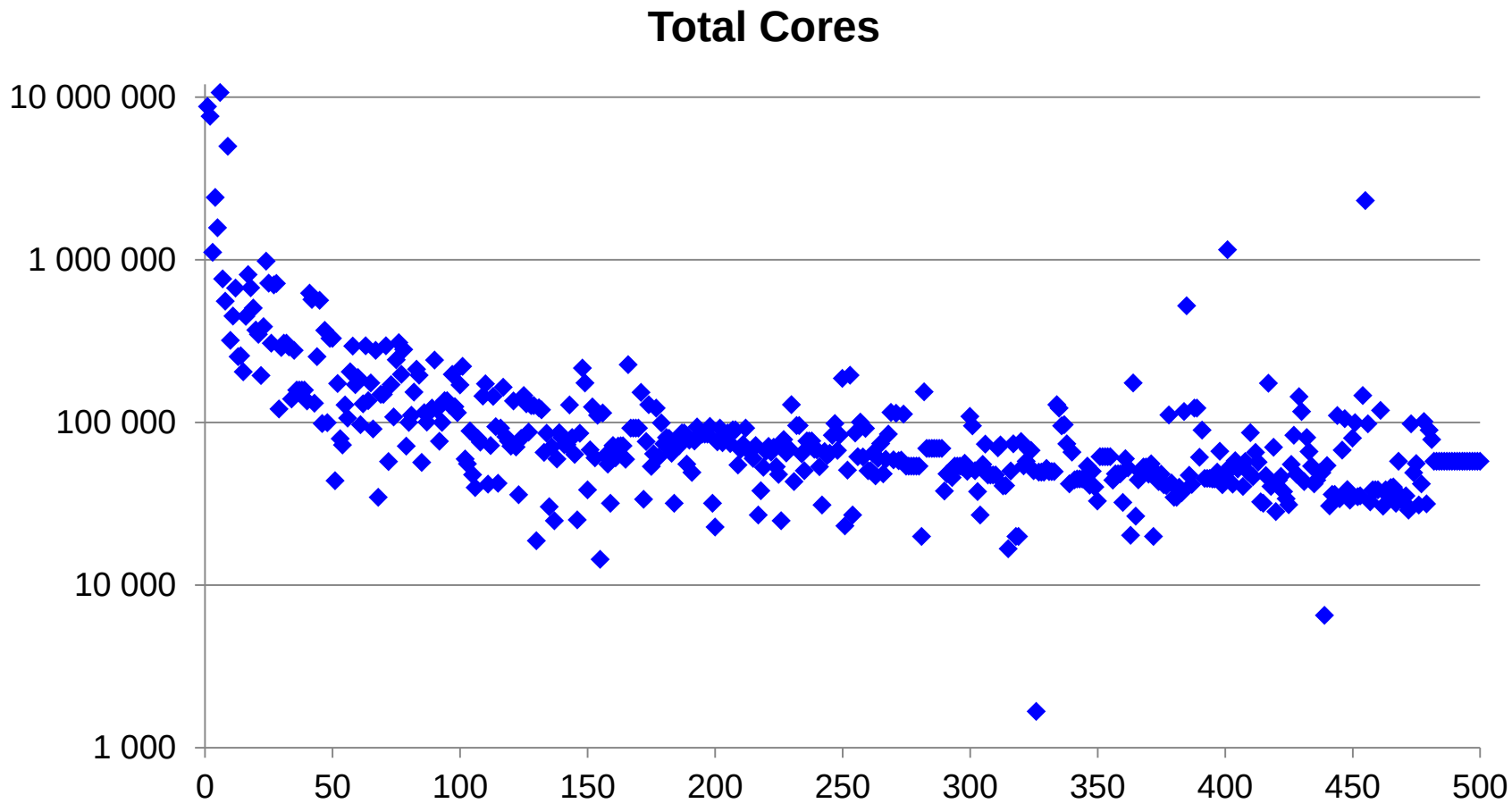
Рациональная декомпозиция сеточных графов

Якобовский Михаил Владимирович

МФТИ, г. Долгопрудный

весна, 2023

Число вычислительных ядер суперкомпьютеров top-500

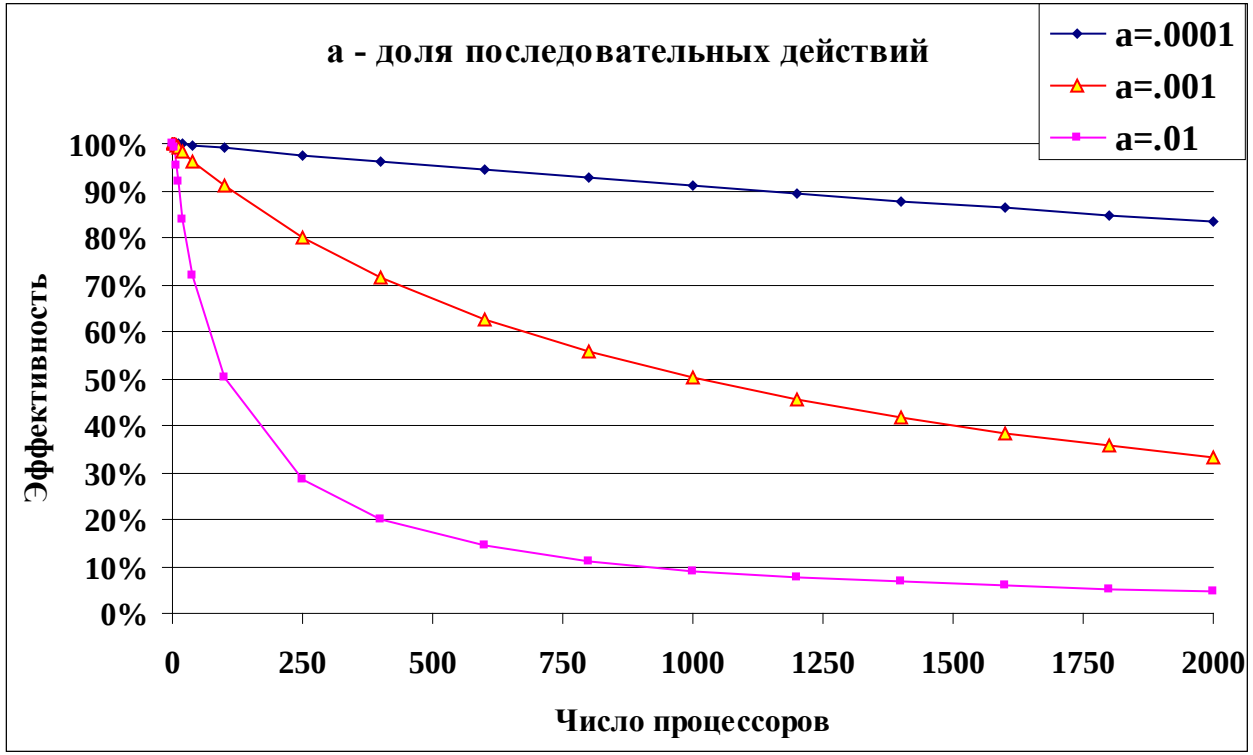


Ограничения

- Закон Амдаля

$$S(p) = \frac{1}{a + \frac{1-a}{p}}$$

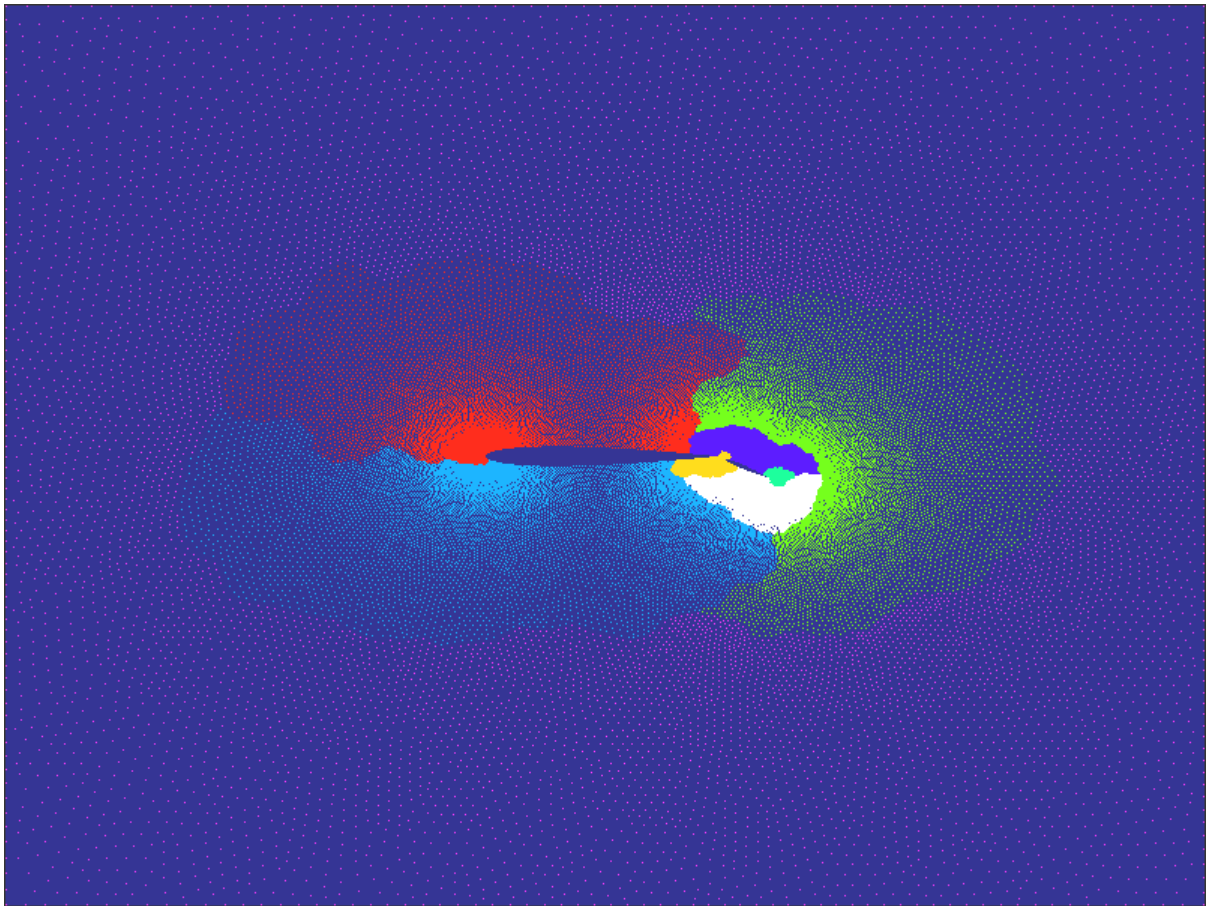
$$E(p) = \frac{1}{1 + a(p-1)}$$

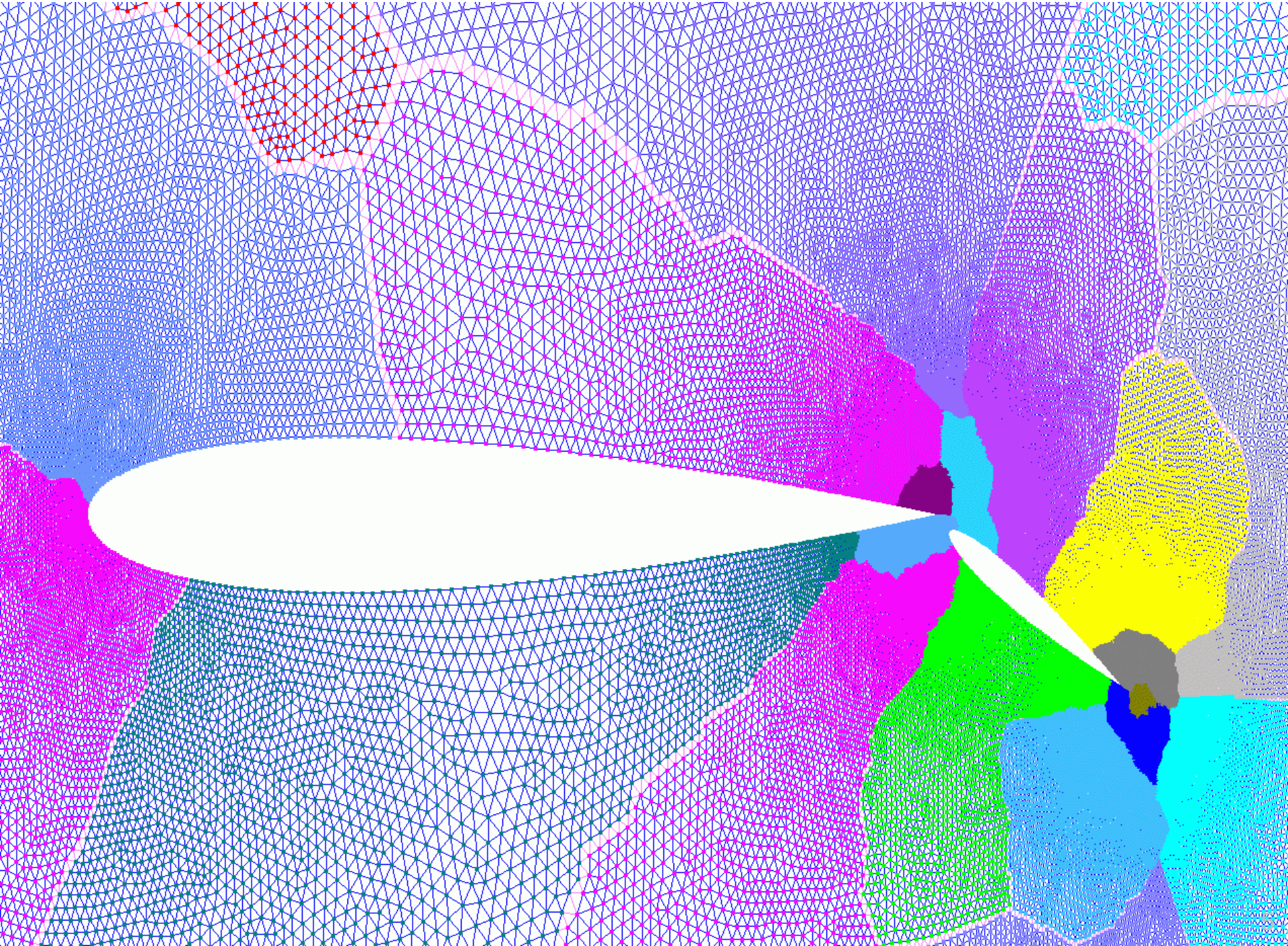


- Пакетный режим исполнения и отладки приложений
- Процедуры авторизованного доступа к удаленным системам
- Высокая динамика изменения конфигурации суперкомпьютеров
- Несоизмеримость ресурсов рабочей станции пользователя и суперкомпьютера

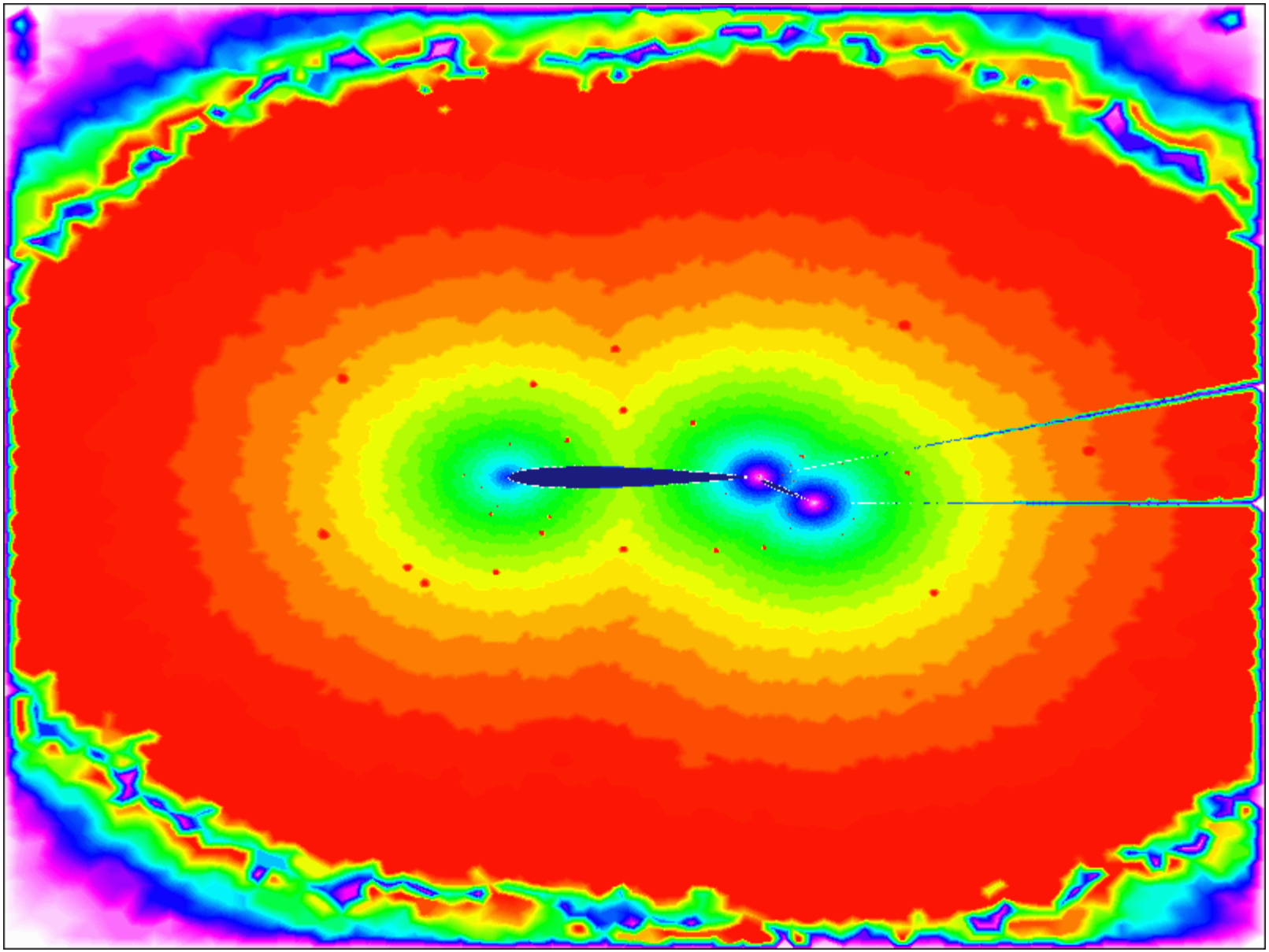
Статическая балансировка загрузки

- Критерии декомпозиции
- Инкрементный алгоритм декомпозиции
- Иерархическая обработка больших сеток

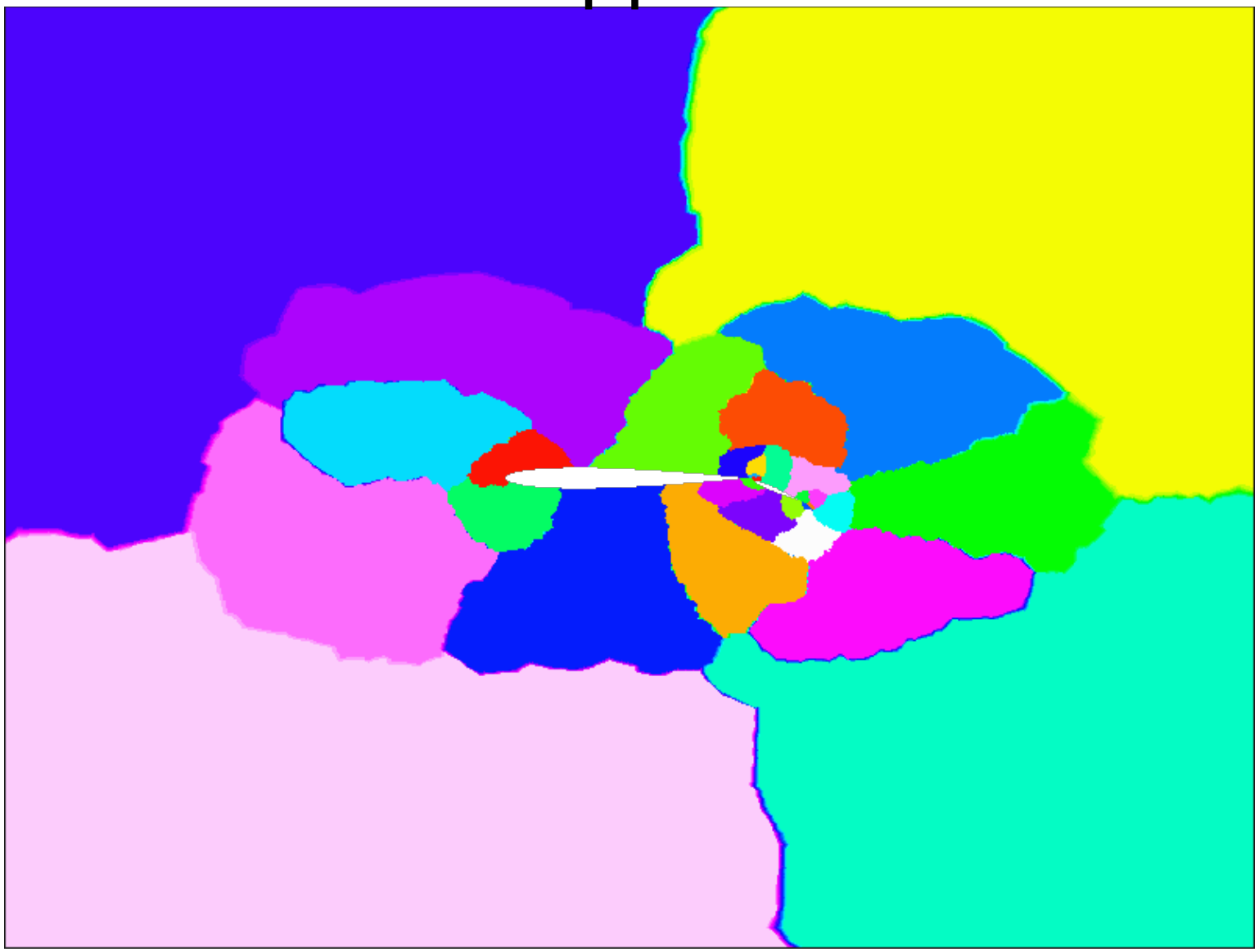




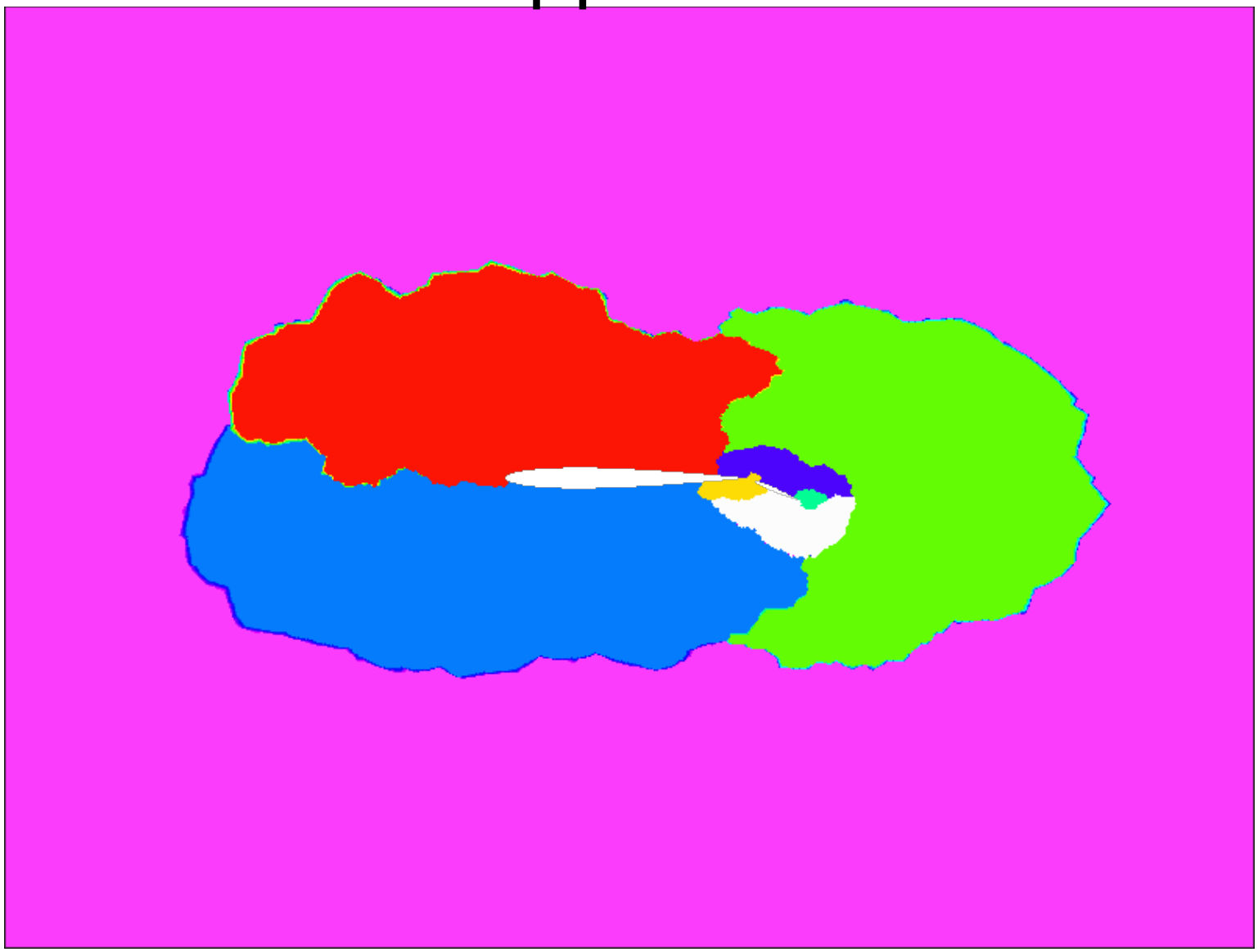
Простое разбиение на 32 домена



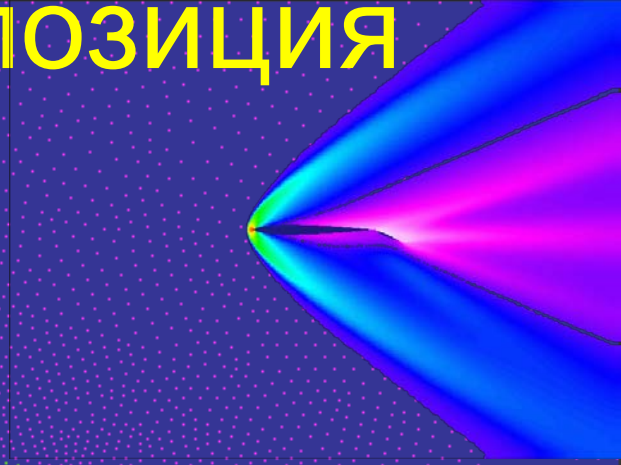
Рациональное разбиение на 32 домена

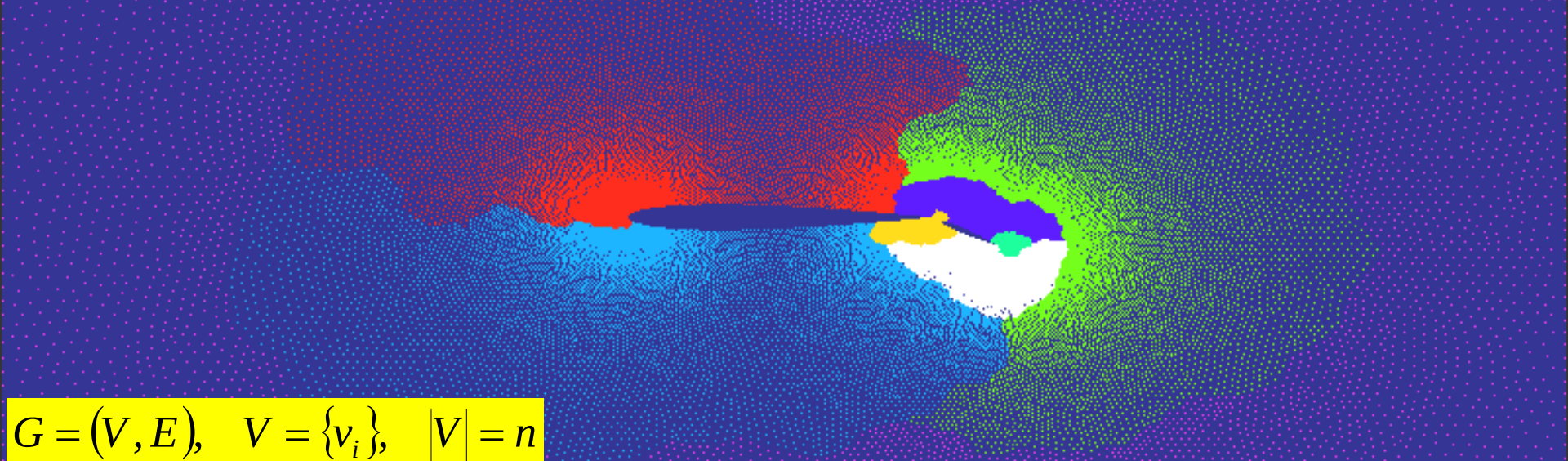


Рациональное разбиение на 8 доменов



Рациональная декомпозиция сеток



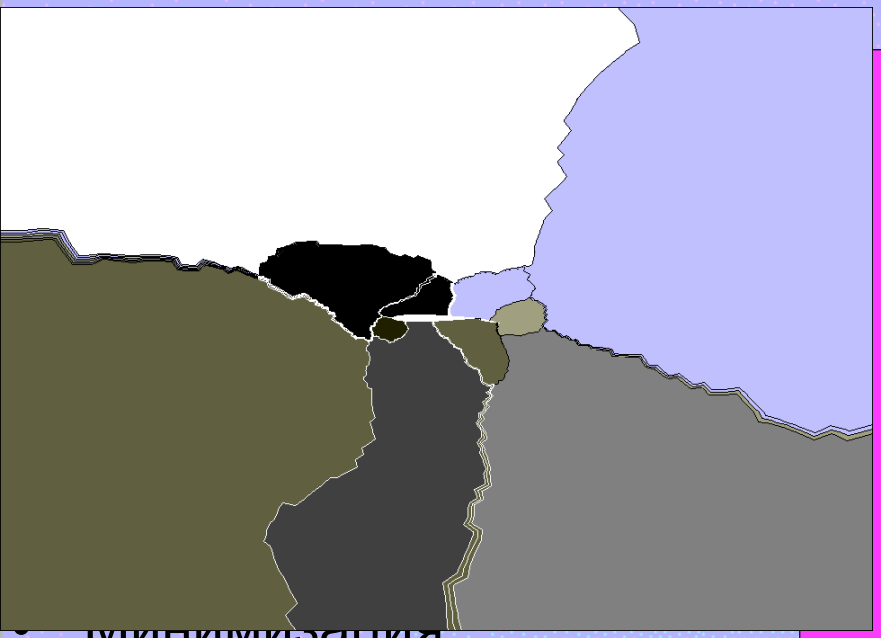


$$G = (V, E), \quad V = \{v_i\}, \quad |V| = n$$

$$V = \bigcup_{k=1}^{p} V_k, \qquad V_i \cap V_j = \emptyset, \text{при } i \neq j$$

$$\min_{R(V)} \left\{ J = \max_{k=1,\ ,p} \sum_{v_i \in V_k} \left(w(v_i) + \alpha \sum_{v_j \notin V_k} w(v_i, v_j) \right) \right\}$$

Критерии декомпозиции графов



- Минимизация максимальной степени доменов
- Обеспечение связности доменов
- Обеспечение связности множества внутренних узлов доменов

А.Н. Андрианов, А.В. Жохова, Б.Н. Четверушкин

Процессоров	11	31	47	63
New_sort	13.59	5.59	4.38	4.16
METIS	13.61	11.00	11.10	10.56

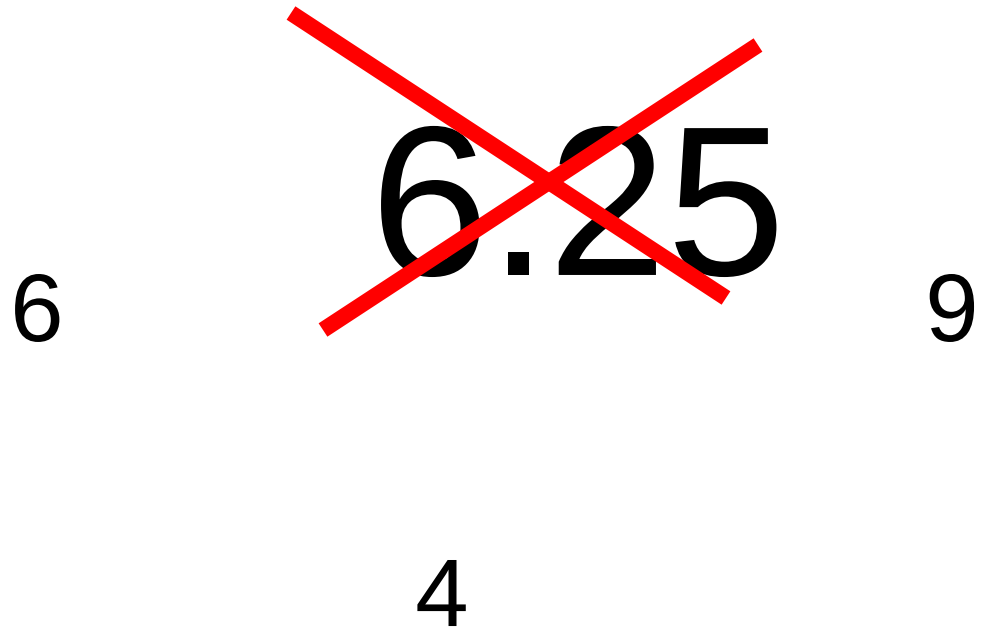
Чему равно 25/4 ?

6.25

25/4=

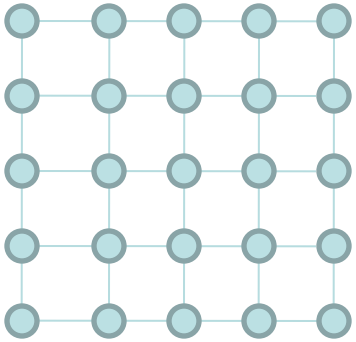


25/4=

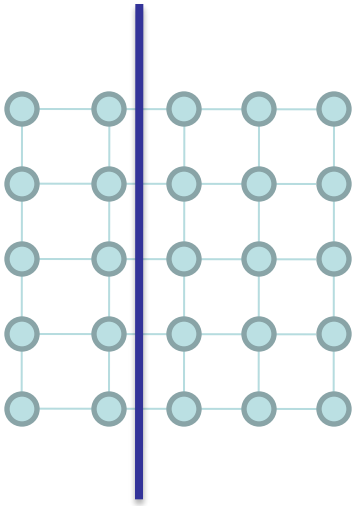


25/4 = 4 ? 6 ? 9

- Разрезать решетку 5 x 5 на 4 части

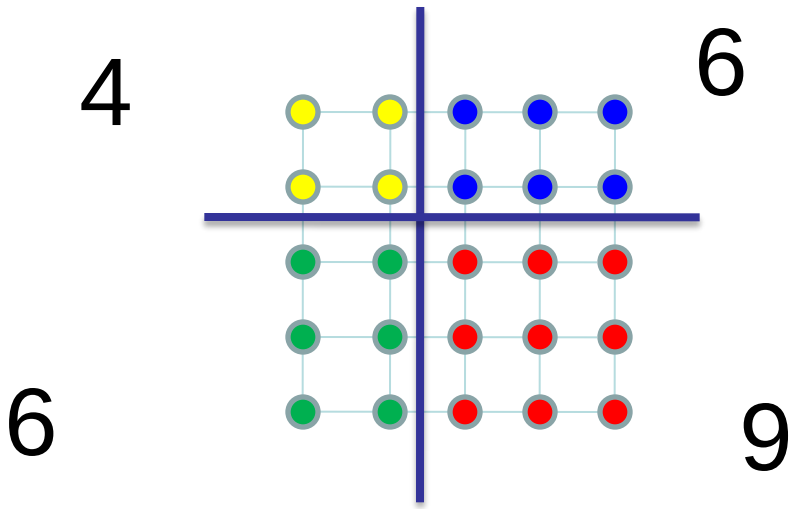


Декомпозиция сетки из 25 узлов на 4 части



25/4 = 4 ? 6 ? 9

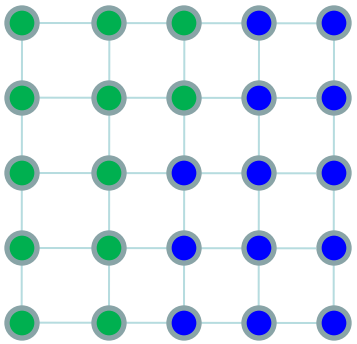
- Декомпозиция решетки 5 x 5 на 4 домена



- Дисбаланс 9/4=2.25

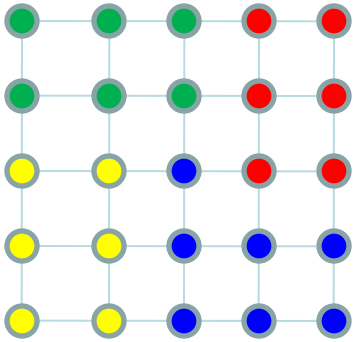
25/4 = 4 ? 6 ? 9

- Декомпозиция решетки 5 x 5 на 2 домена
- Дисбаланс 13/12 : 8%



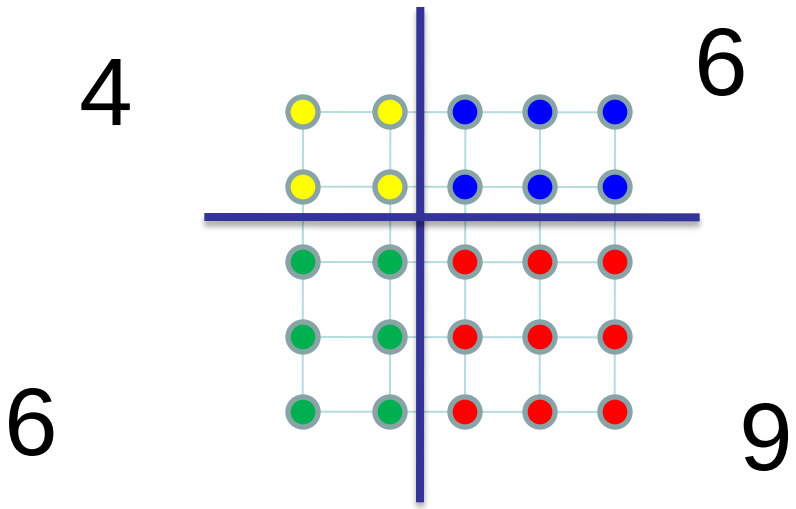
25/4 = 4 ? 6 ? 9

- Декомпозиция решетки 5 x 5 на 4 домена
- Дисбаланс 7/6 : 17%



25/4 = 4 ? 6 ? 9

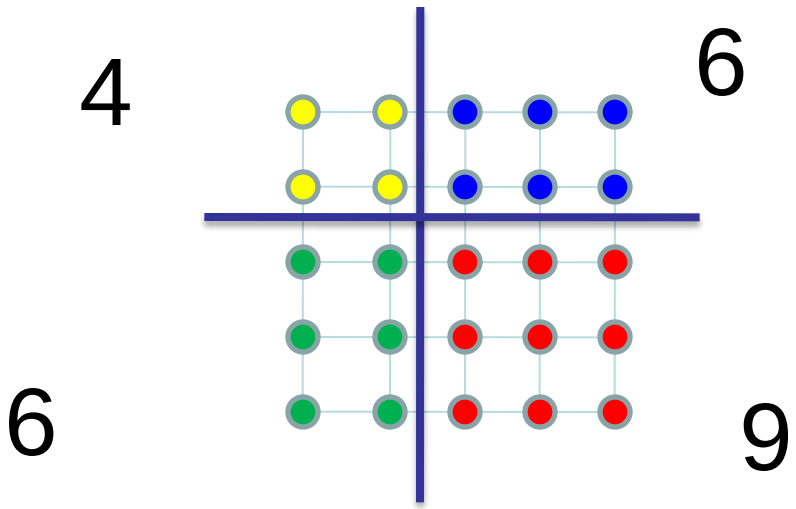
- Декомпозиция решетки 5 x 5 на 4 домена



- Дисбаланс 9/4=2.25

25/4 = 4 ? 6 ? 9

- Декомпозиция решетки 5 x 5 на 4 домена

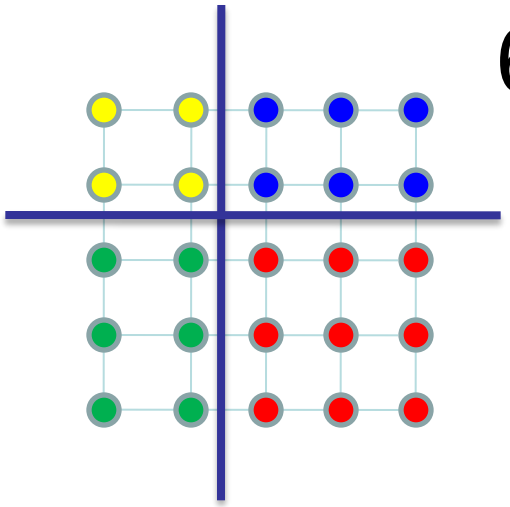


- ~~Дисбаланс 9/4=2.25~~

25/4 = 4 ? 6 ? 9

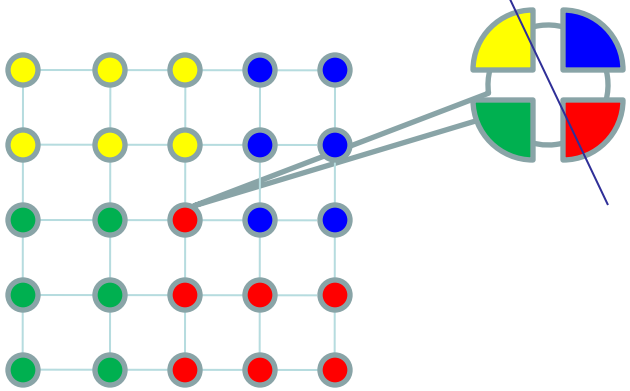
- Декомпозиция решетки 5 x 5 на 4 домена

4 6



6 9

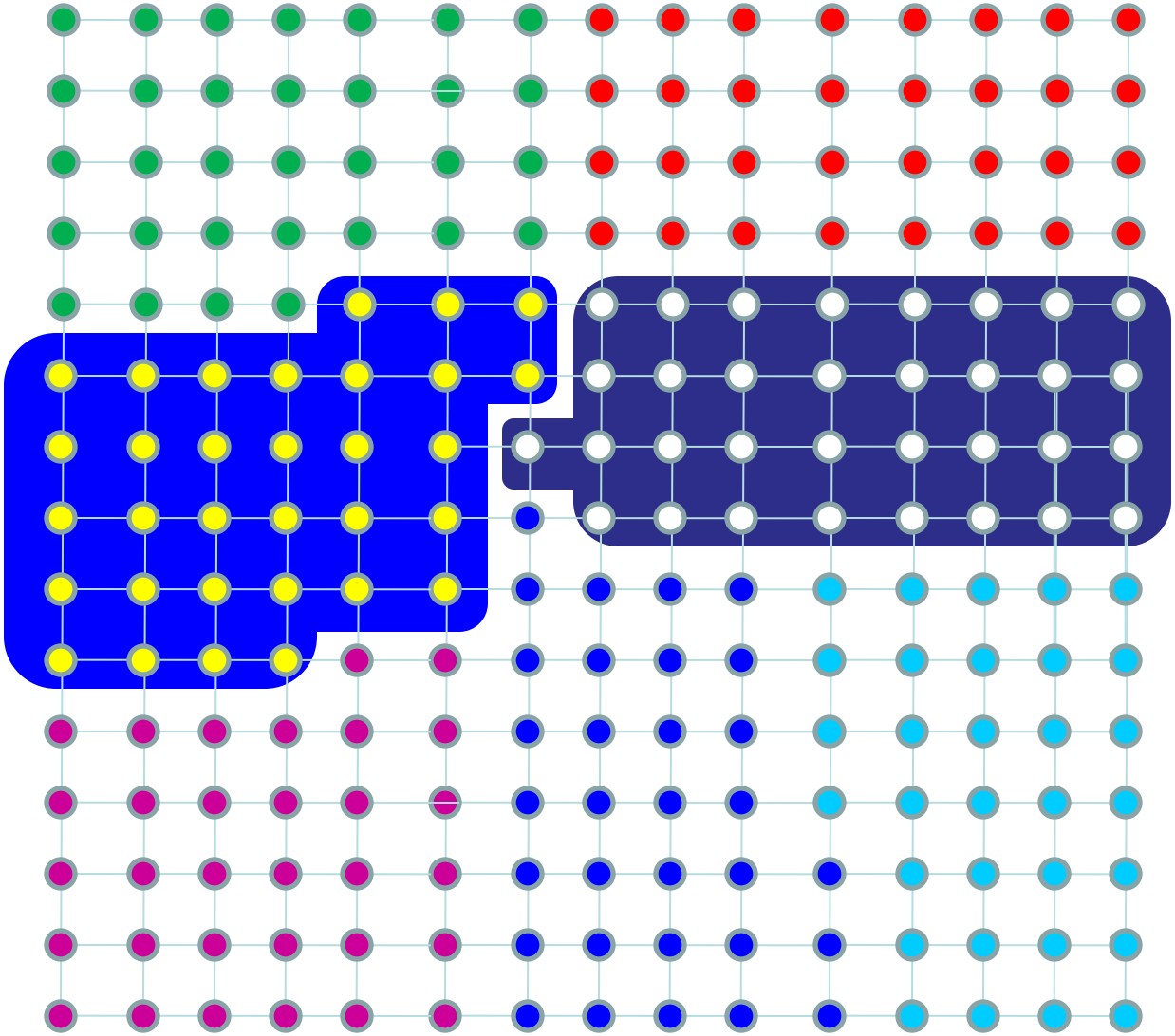
Потери
9/6.25=1.44

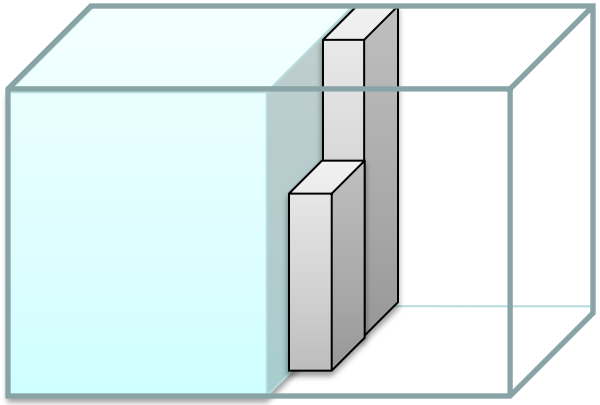


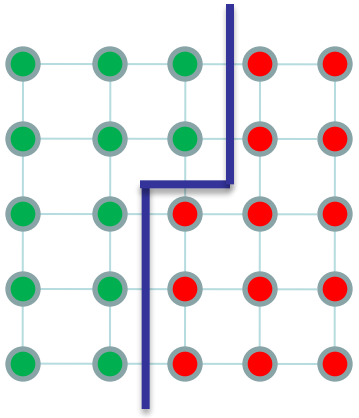
Потери
9/7=1.29

- ~~Дисбаланс 9/4=2.25~~

Декомпозиция сетки 25x25 на 7 частей







Разрезано ребер

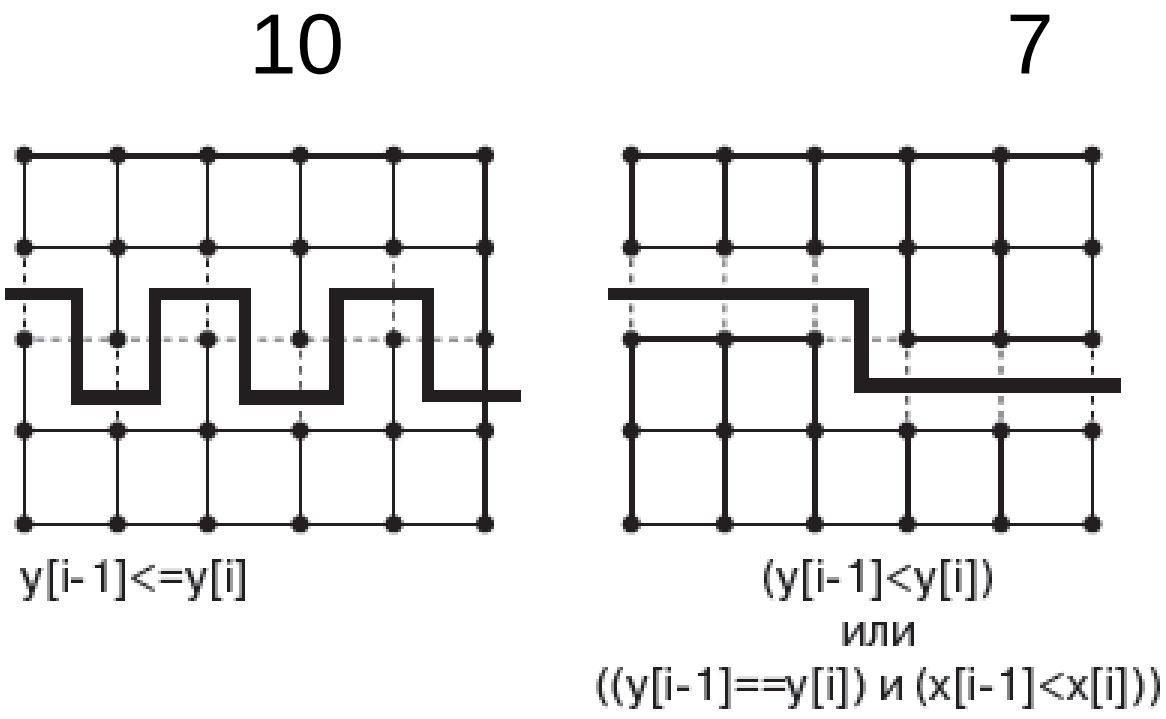


Рис. 76. Координатная бисекция

Последовательные пакеты декомпозиции сеток

METIS, Jostle, Scotch, Chaco, Party

Параллельные пакеты декомпозиции сеток

ParMETIS, Jostle, PT-Scotch, Zoltan

- **нерегулярные сетки, содержащие 10^9 и более вершин**

Пакеты декомпозиции графов

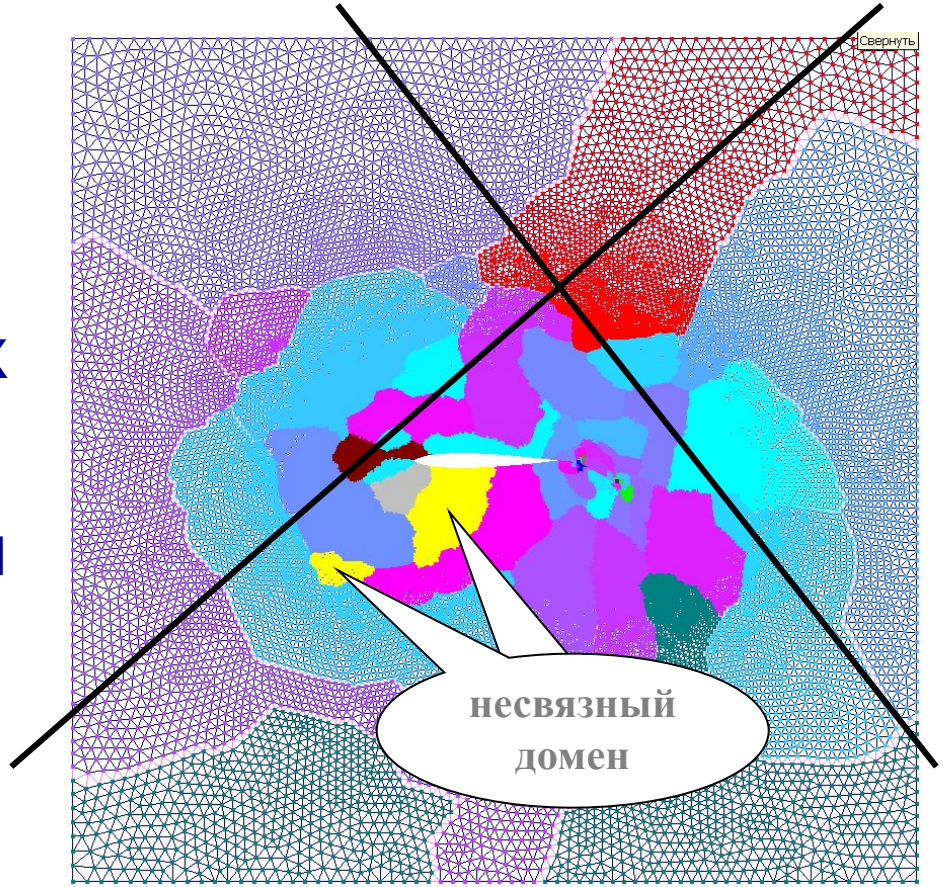
Chaco	Bruce Hendrickson Robert Leland
ParMETIS	George Karypis Vipin Kumar
PARTY	Robert Prais, et al.
JOSTLE	Chris Walshaw, et al.
SCOTCH	Francois Pellegrini

Недостатки алгоритмов декомпозиции графов, реализованных в существующих пакетах

- образование несвязных доменов
- формирование сильно несбалансированных разбиений (ParMETIS: количества вершин в доменах могут отличаться в 2 раза)
- не всегда удается получить разбиения на большое число микродоменов

Связность важна:

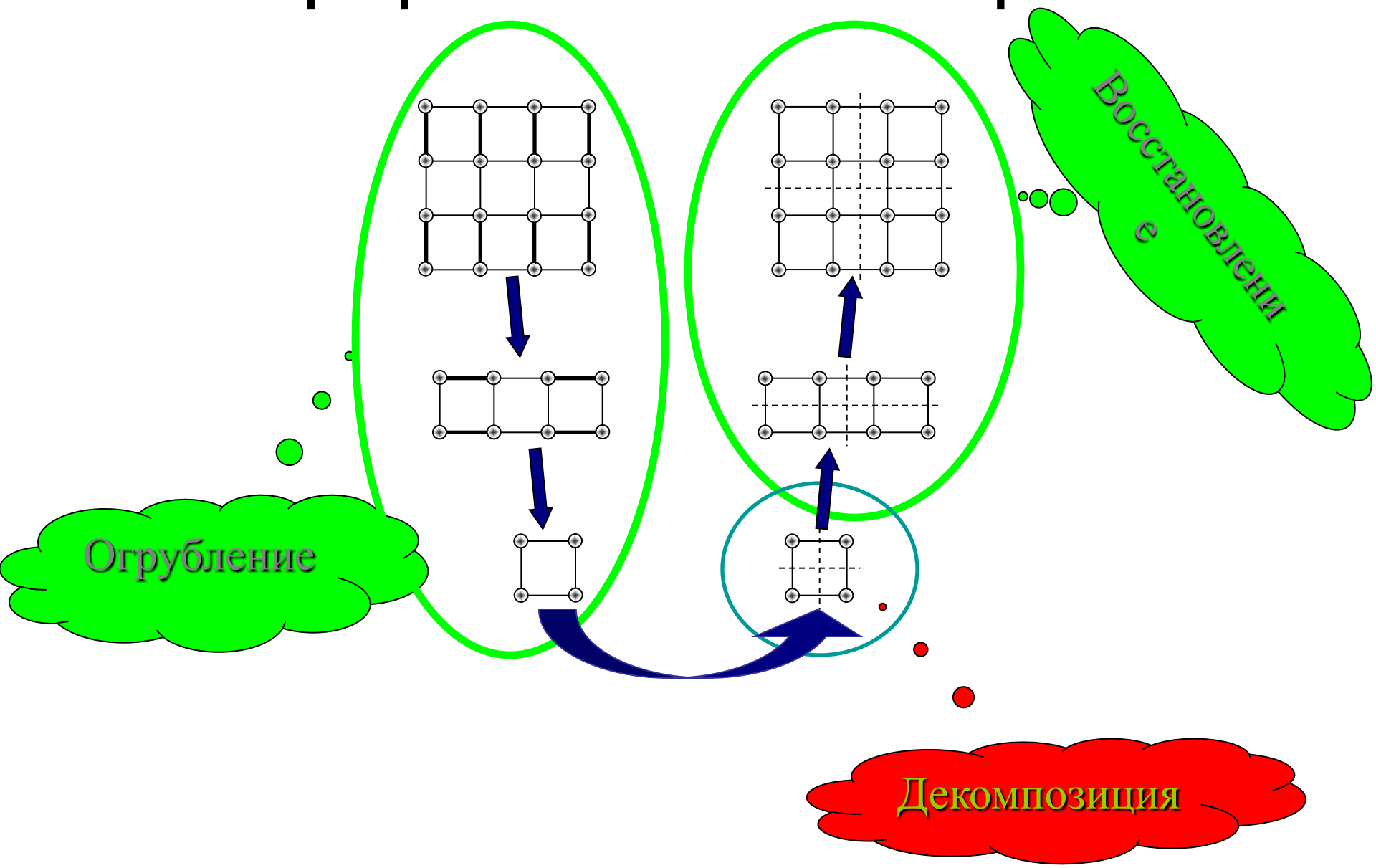
- **алгоритмы решения систем линейных уравнений**
- **компрессия сеточных данных**
- **алгоритм композиции подобластей[1]**
- **распараллеливание[2] методики ТИМ-2D**



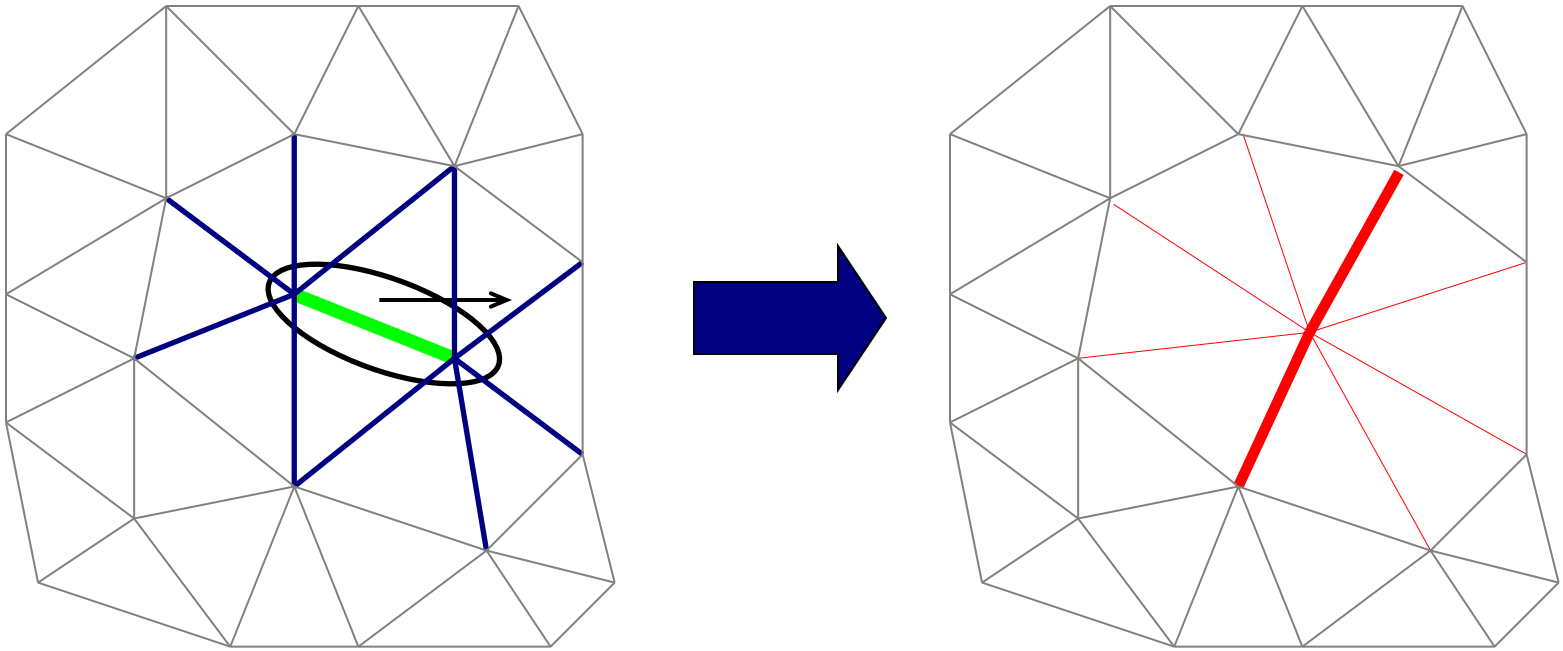
[1] А. И. Илюшин, А. А. Колмаков, И. С. Меньшов. Построение параллельной вычислительной модели путем композиции вычислительных объектов // Математическое моделирование. 2011. Т. 23. № 7. 97-113.

[2] А. А. Воропинов. Декомпозиция данных для распараллеливания методики ТИМ-2D и критерии оценки ее качества // Вестник ЮУрГУ. Серия «Математическое моделирование и программирование:», вып. 4. 2009. №37(170). 40-50.

Иерархический алгоритм



Огрубление графа



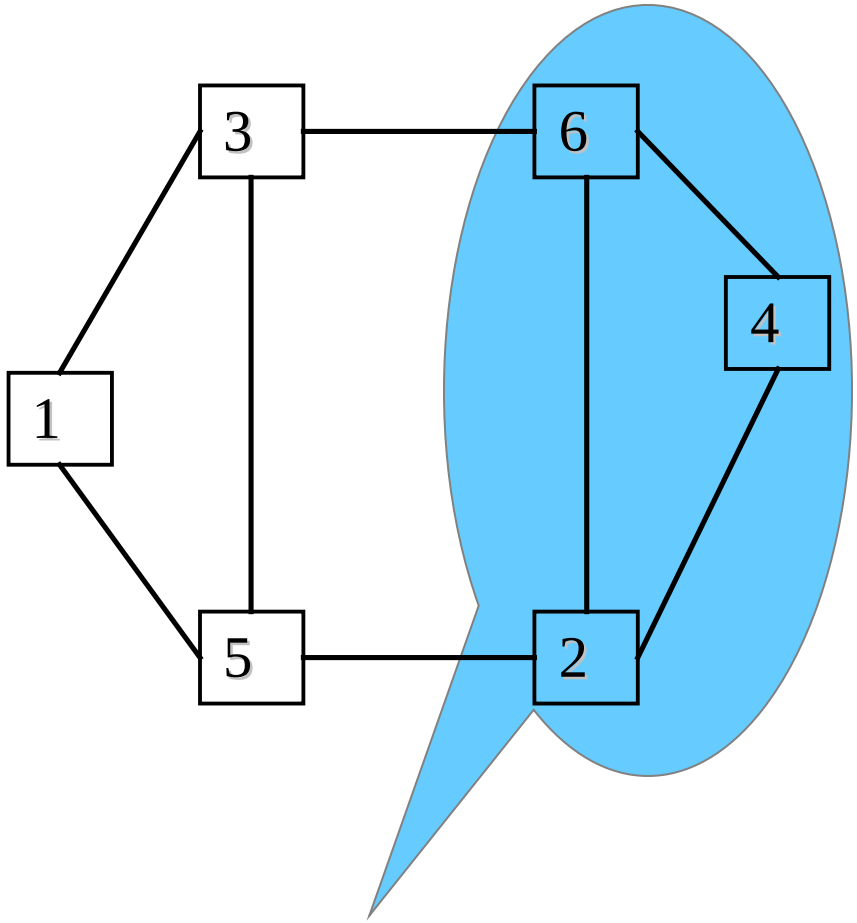
Спектральный метод

$$A = \begin{pmatrix} 2 & 0 & -1 & 0 & -1 & 0 \\ 0 & 3 & 0 & -1 & -1 & -1 \\ -1 & 0 & 3 & 0 & -1 & -1 \\ 0 & -1 & 0 & 2 & 0 & -1 \\ -1 & -1 & -1 & 0 & 3 & 0 \\ 0 & -1 & -1 & -1 & 0 & 3 \end{pmatrix}$$

$$\lambda_1 = 1$$

$$\lambda_{0-5} = \{0,1,3,3,4,5\}$$

$$x_1 = (2, -1, 1, -2, 1, -1)$$



$$\mathbf{s} = (4, 2, 6, 3, 5, 1)$$

Спектральная бисекция

Пусть $q[i]=\{-1,1\}$ определяет принадлежность вершины i одному из двух доменов

Тогда, для равенства числа вершин в доменах, необходимо

$$\sum_{i=1}^{|V|} q_i = 0$$

Спектральная бисекция

Общее число вершин $$\sum_{i=1}^{|V|} q_i^2 = q^T q = |V|$$

Матрица Лапласа $$l_{ij} = \begin{cases} -1, & e_{ij} \in E,\ i \neq j, \\ \sum_{k \neq i} -l_{ik}, & i = j, \\ 0, & \text{иначе} \end{cases}$$

Число разрезанных ребер

$$|E_c| = \frac{1}{4} \sum_{i,j:e_{i,j} \in E} \left(q_i - q_j\right)^2$$

$$Lq = \lambda q$$

$$q^T L q = \sum_{e_{i,j} \in E} \left(q_i - q_j\right)^2 = q^T \lambda q = \lambda |V|$$

Следовательно $|E_c| = \frac{1}{4} \lambda |V|$

Число разрезанных ребер

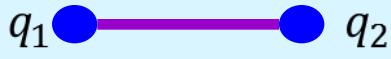
$$|E_c| = \frac{1}{4} \sum_{i,j:e_{i,j} \in E} (q_i - q_j)^2$$

$$Lq = \lambda q$$

$$(q_1, q_2) \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} =$$

$$q_1^2 - 2q_1q_2 + q_2^2 =$$

$$(q_1 - q_2)^2$$



$$q^T Lq = \sum_{e_{i,j} \in E} (q_i - q_j)^2 = q^T \lambda q = \lambda |V|$$

Следовательно $|E_c| = \frac{1}{4} \lambda |V|$

Разбиение вершин на два множества

Для минимизации $|E_c| = \frac{1}{4}\lambda|V|$ $\lambda_0 = 0$ следует найти минимальное собственное число и соответствующий ему собственный вектор – вектор Фидлера ψ

Он ортогонален вектору соответствующему нулевому λ_0 - единичному вектору $L\psi = \lambda_1\psi$, следовательно $e\psi = 0, \sum_{i=1}^{|V|}\psi_i = 0,$

значит множества {-1} и {1} содержат одинаковое число элементов

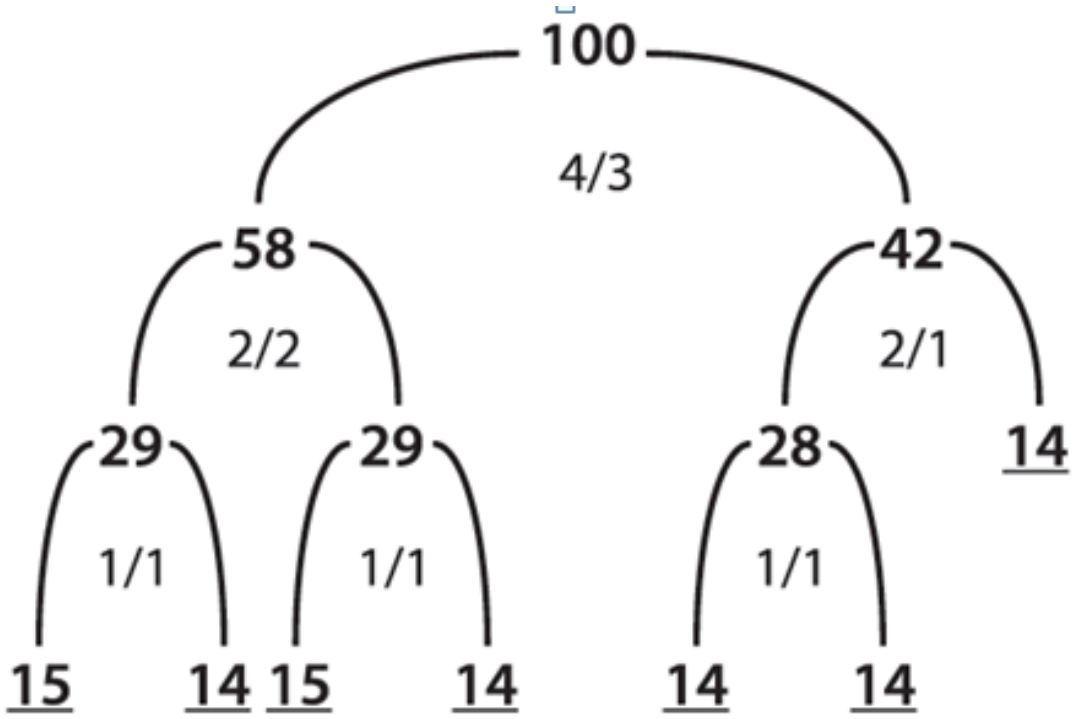
Рекурсивная бисекция


```
RecursiveBisect(graf,n,k)
{
k1=(k+1)/2
k2=k-k1
n1=n*k1/k
n2=n-n1

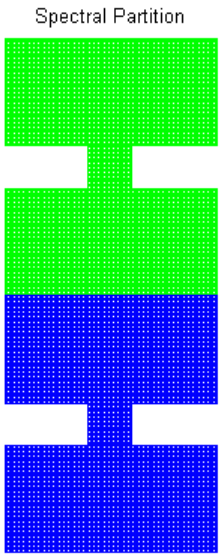
// разбиение вершин графа на две части
// размером n1 и n2
Bisect(graf,n,subgraf1,n1,subgraf2,n2)

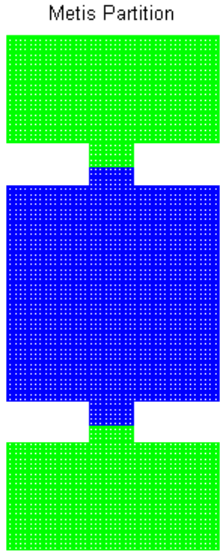
if(k1>1) RecursiveBisect(subgraf1,n1,k1)
if(k2>1) RecursiveBisect(subgraf2,n2,k2)
}
```

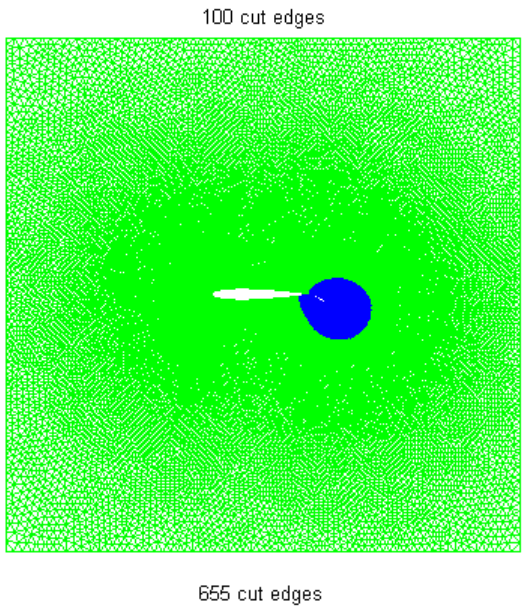
Бинарное дерево разрезов

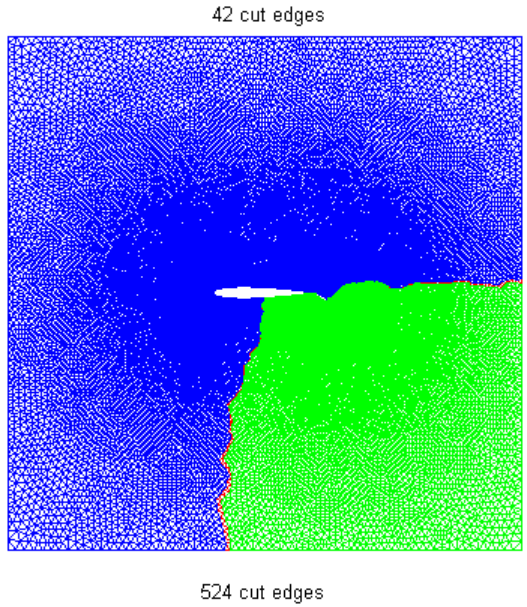


Метод спектральной бисекции

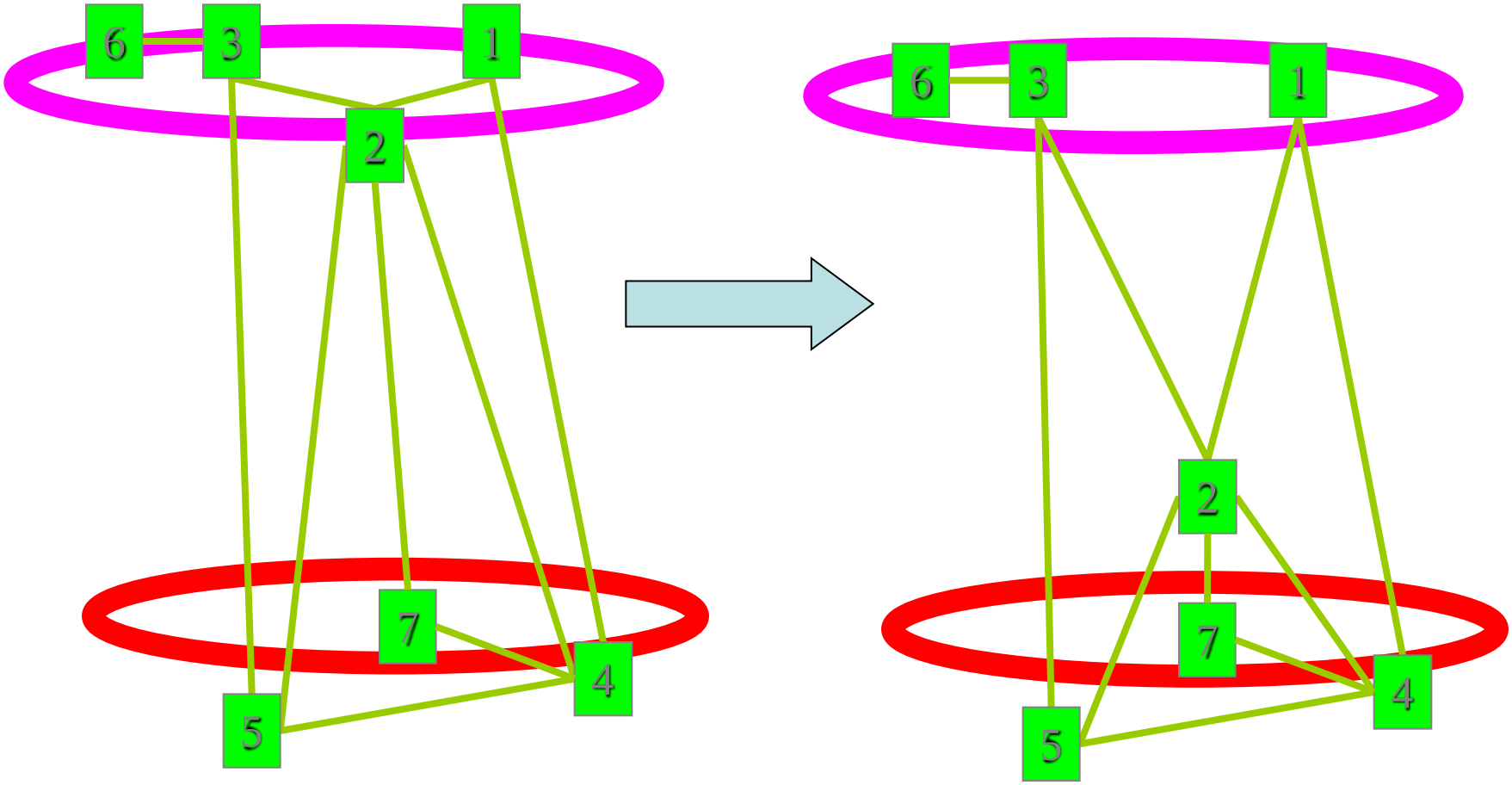








Локальное уточнение

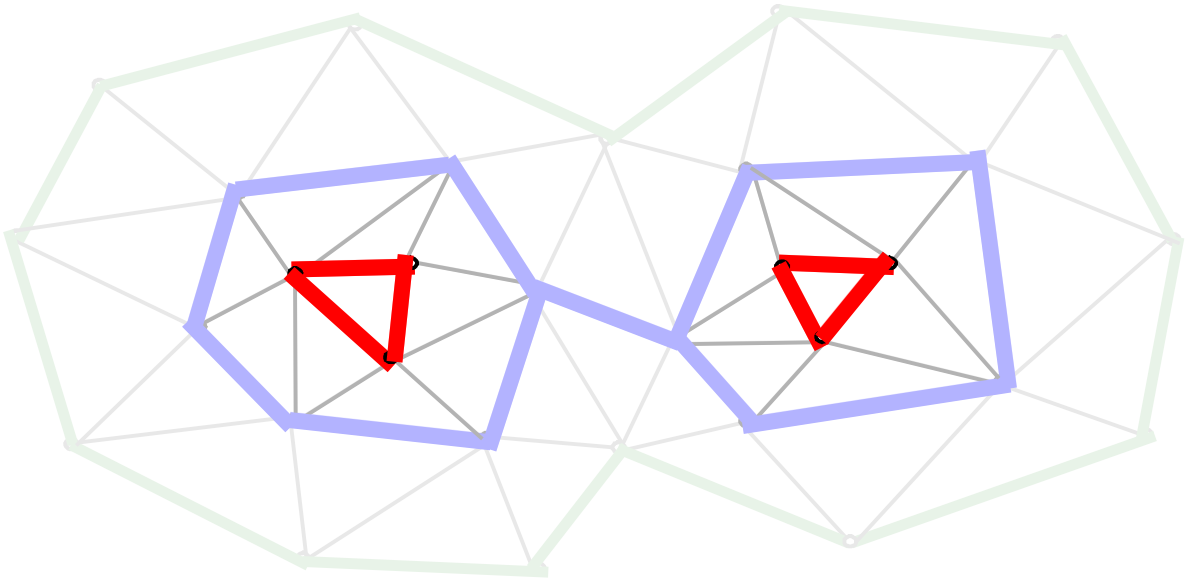


Kernighan-Lin (KL)
и Fiduccia-Mattheyses (FM)

Инкрементный алгоритм декомпозиции графа

1. инициализация доменов
2. распределение вершин по доменам методом инкрементного роста
3. локальное уточнение границ сформированных доменов
4. анализ связности ядер сформированных доменов и окончание работы, если заданный уровень связности достигнут
5. перенос части закрепленных за доменами вершин в группу свободных вершин и переход к этапу 2.

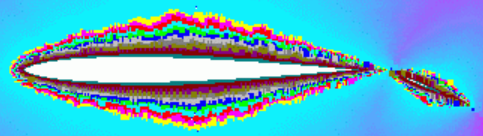
Связность ядер доменов



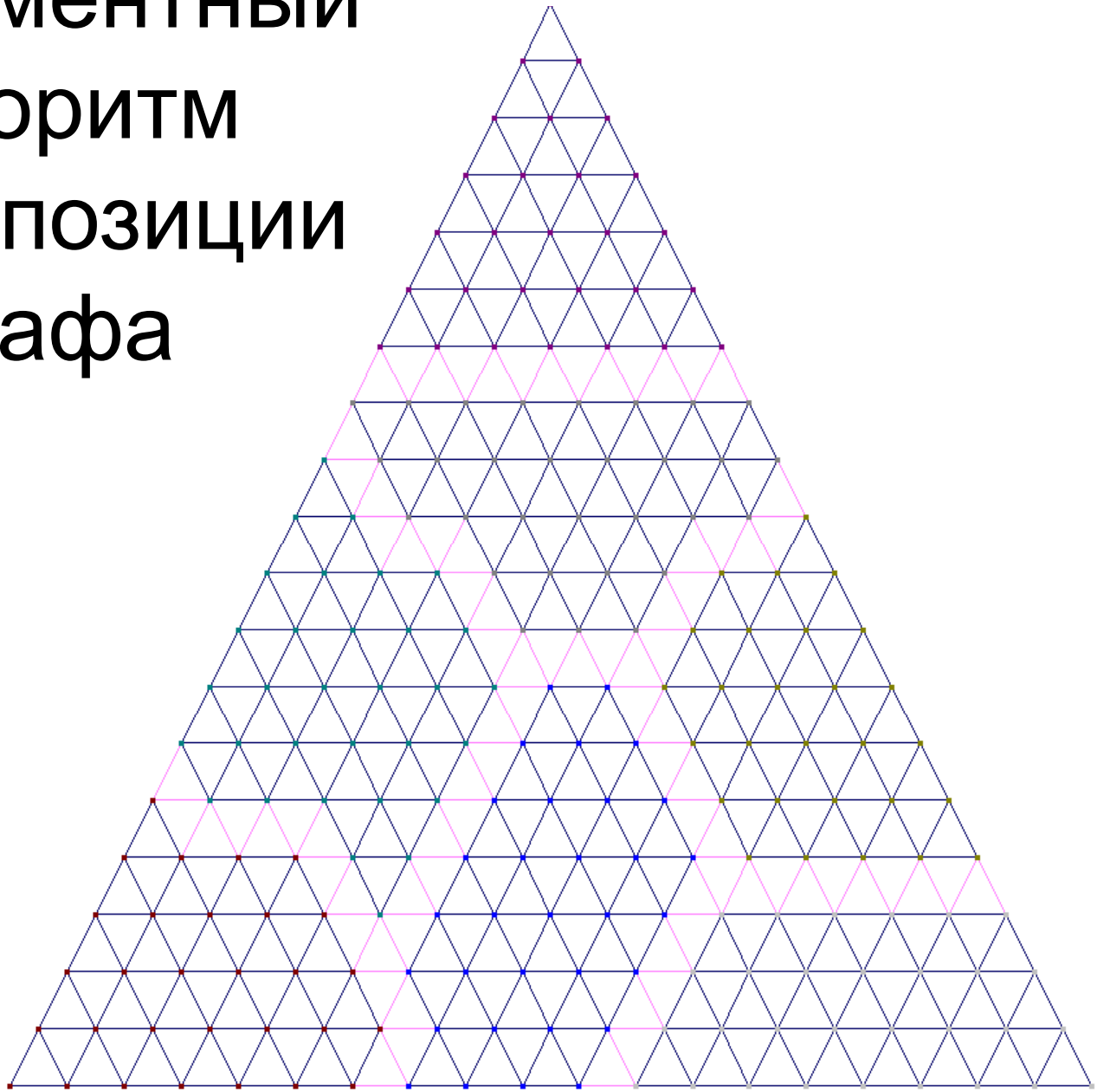
$$d(i) = \min_k : \mathbf{A}^k v_i \cap B \neq \phi$$

$$T_{k+1} = \mathbf{A}T_k \setminus T_k \setminus T_{k-1}, \quad T_{-1} = \phi$$

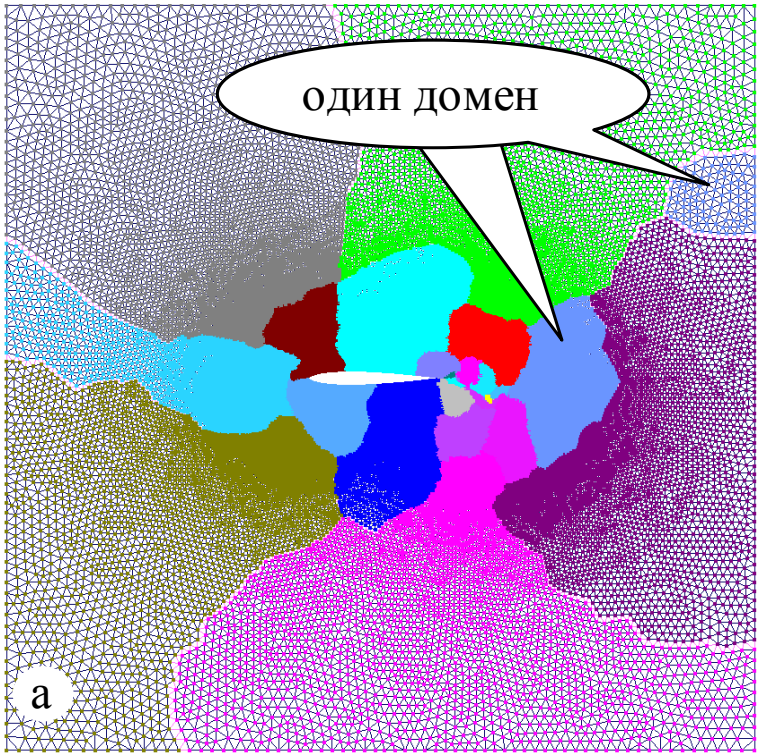
Оболочки

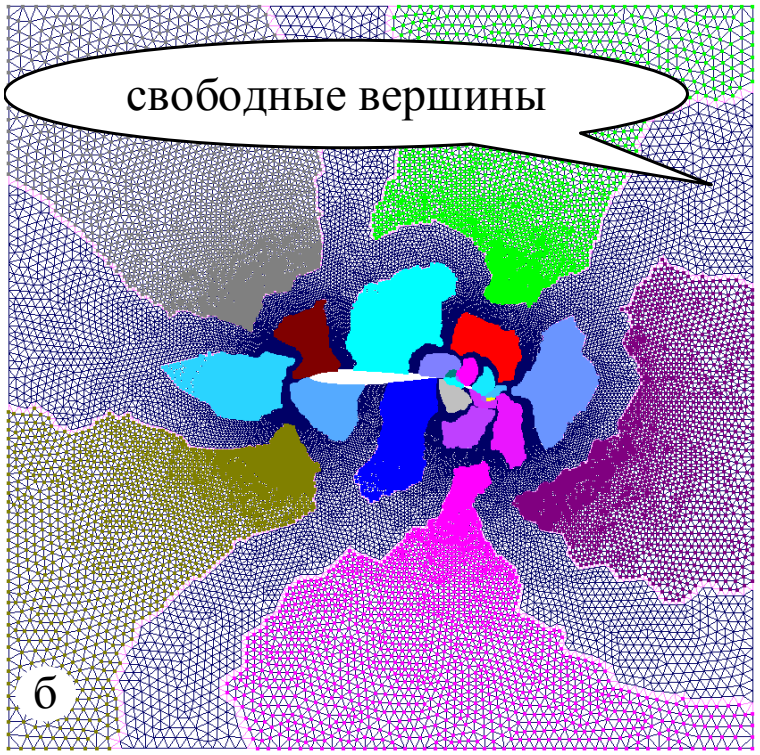


Инкрементный алгоритм декомпозиции графа

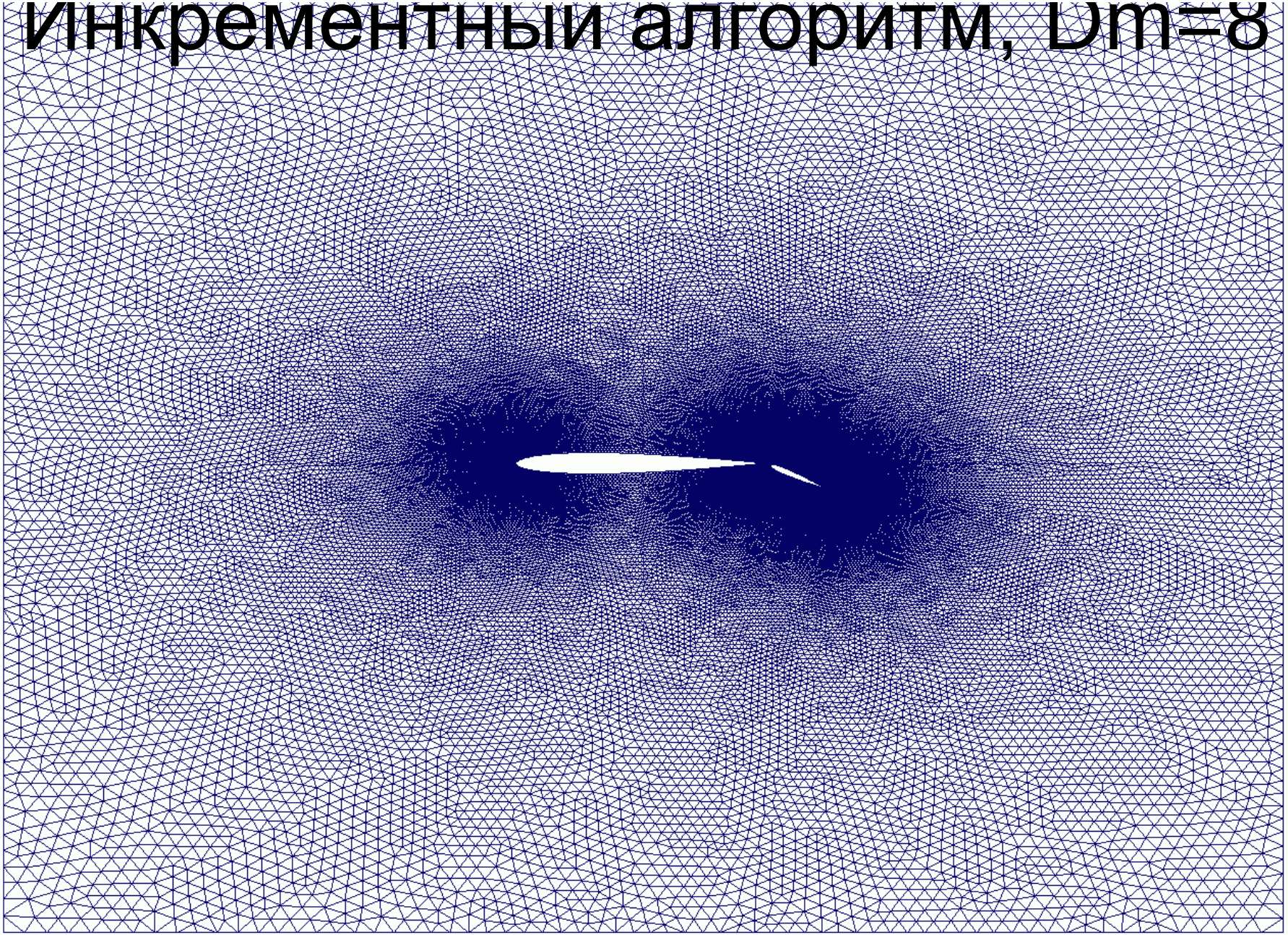


Редуцирование доменов

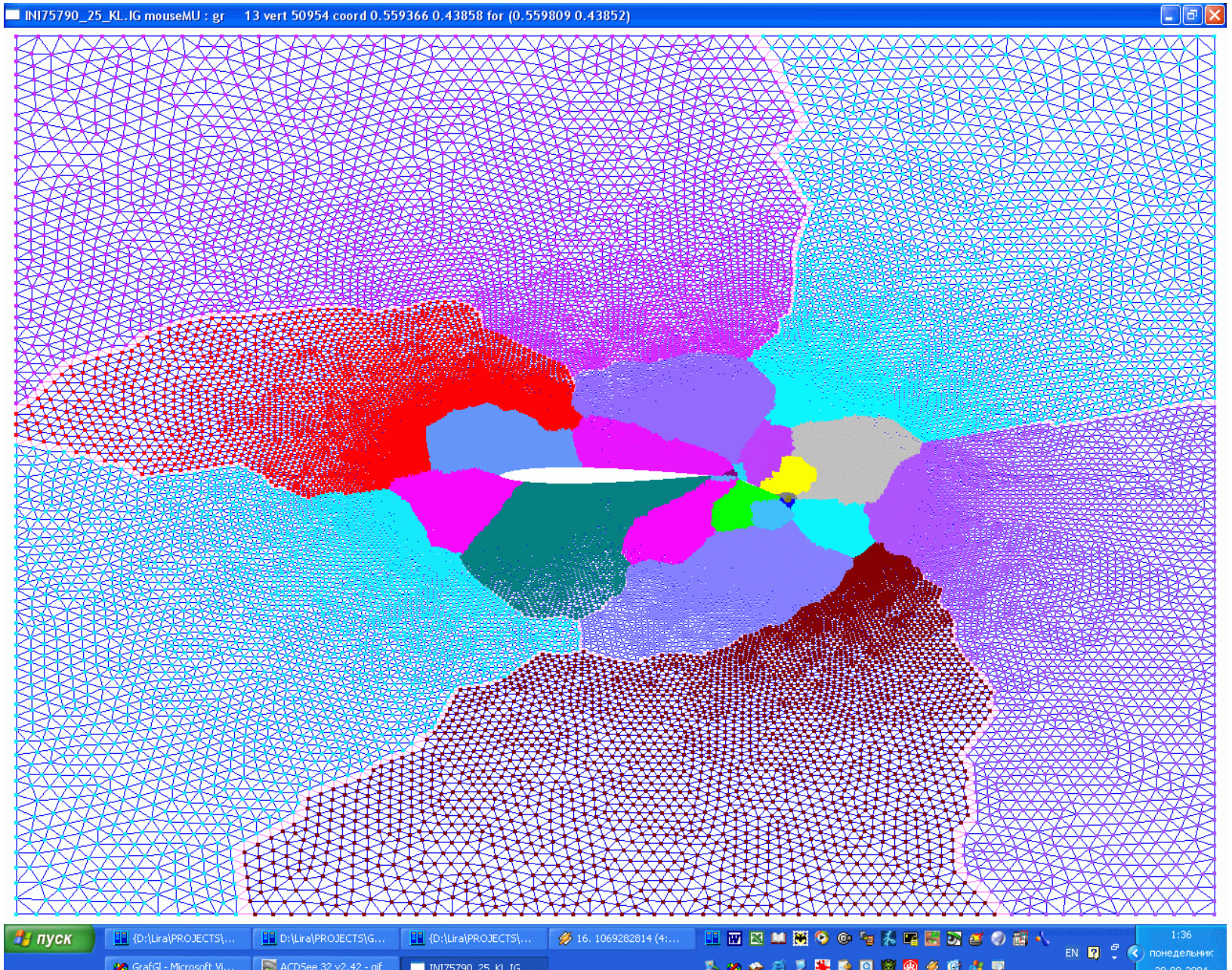




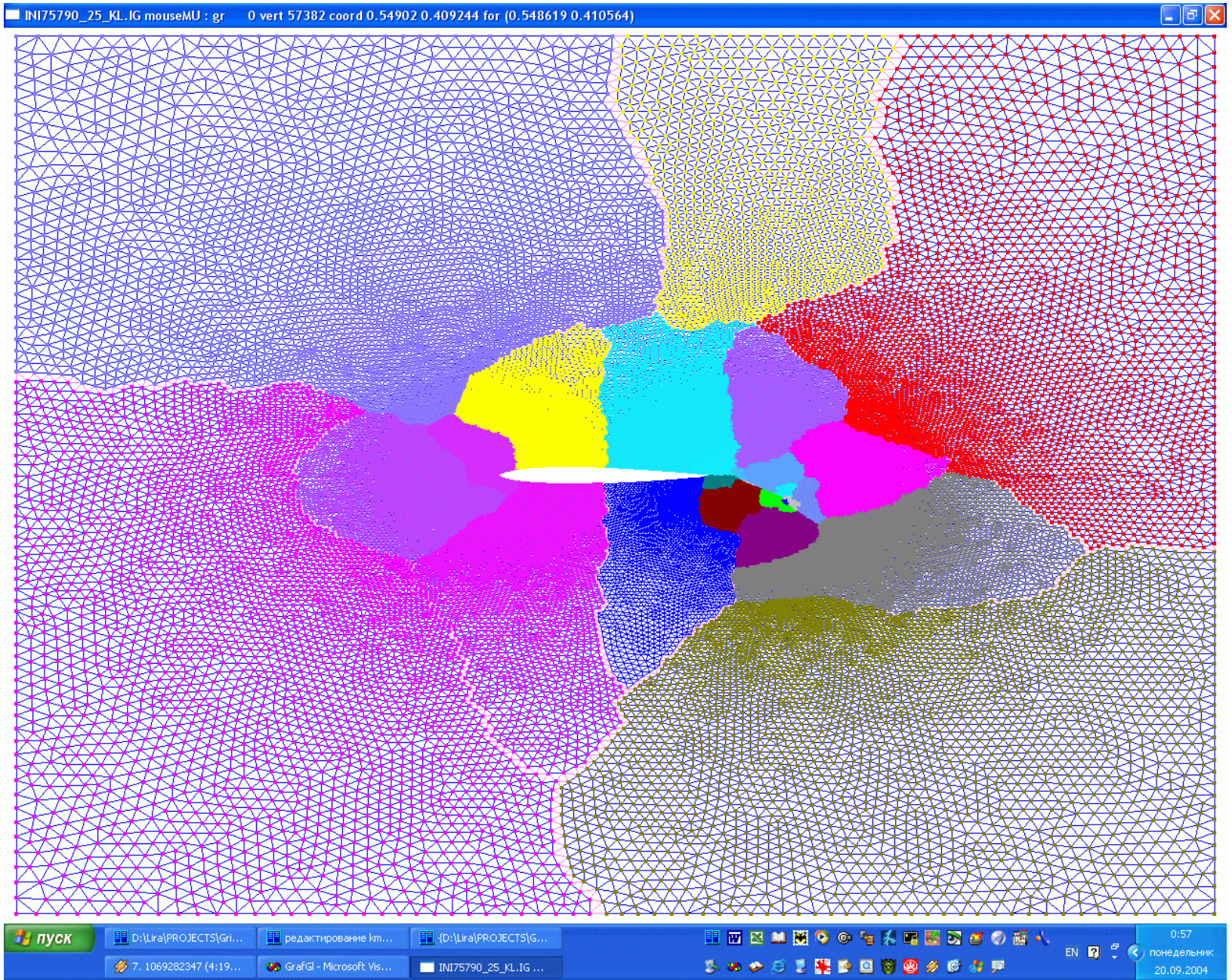
Инкрементный алгоритм, Dm=8



Инкрементный алгоритм, Dm=25

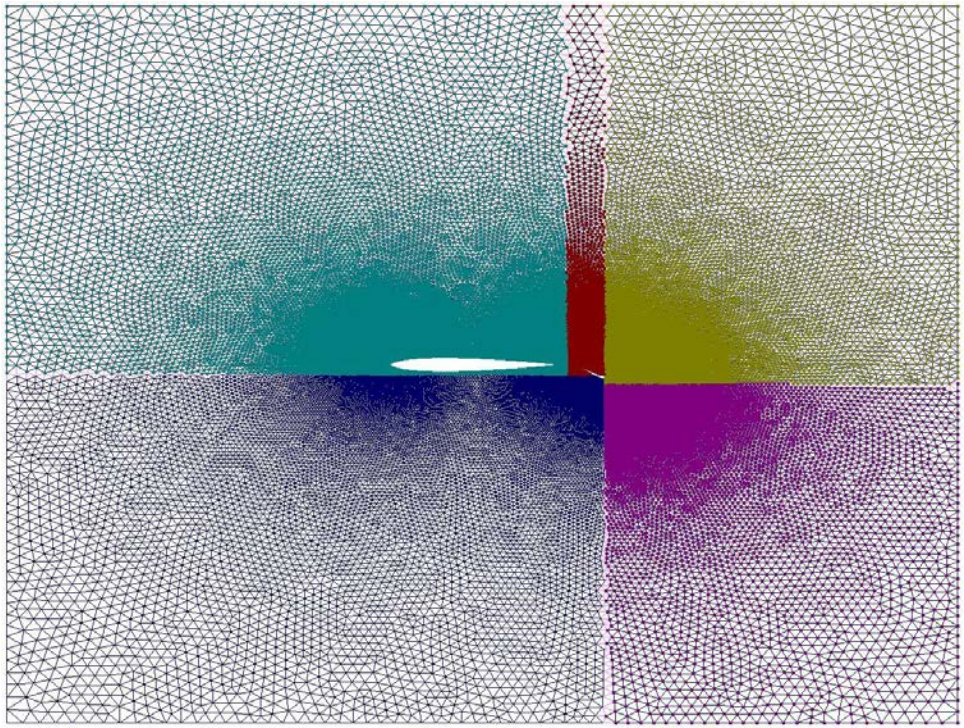


Kmetis, Dm=25

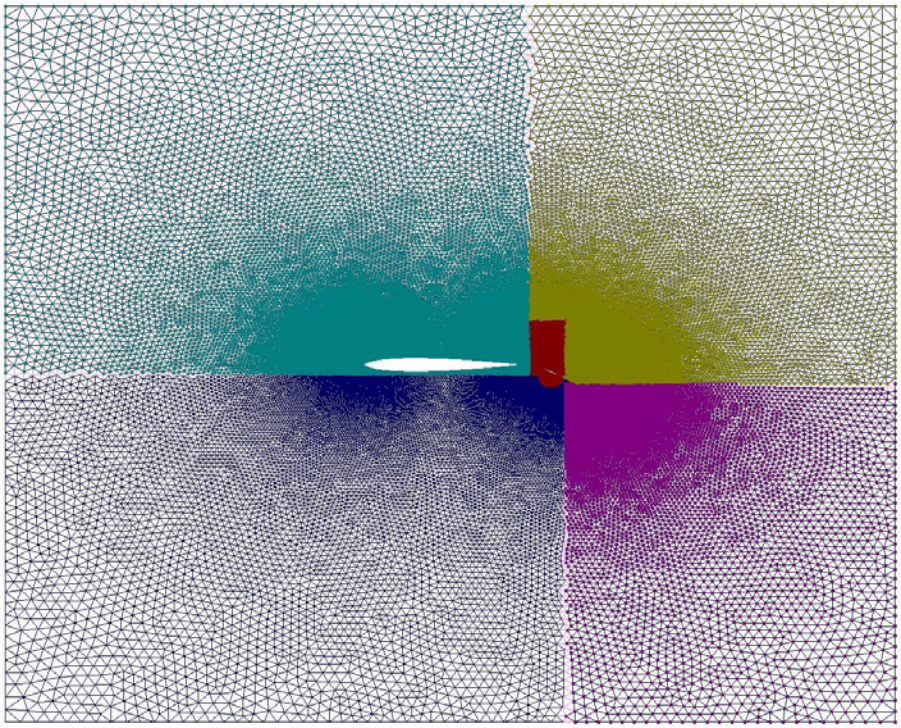

INI75790_25_KL.IG mouseMU : gr 0 vert 57382 coord 0.54902 0.409244 for (0.548619 0.410564)
пуск
D:\Lira\PROJECTS\Gri...
редактирование km...
{D:\Lira\PROJECTS\G...
7. 1069282347 (4:19...
GrafGl - Microsoft Vis...
INI75790_25_KL.IG ...
EN
0:57
понедельник
20.09.2004

Треугольная сетка из 75790 вершин (пространство вокруг крыла)

результат геометрической декомпозиции на 5 групп
(в дальнейшем каждый процессор считывает свою группу вершин)

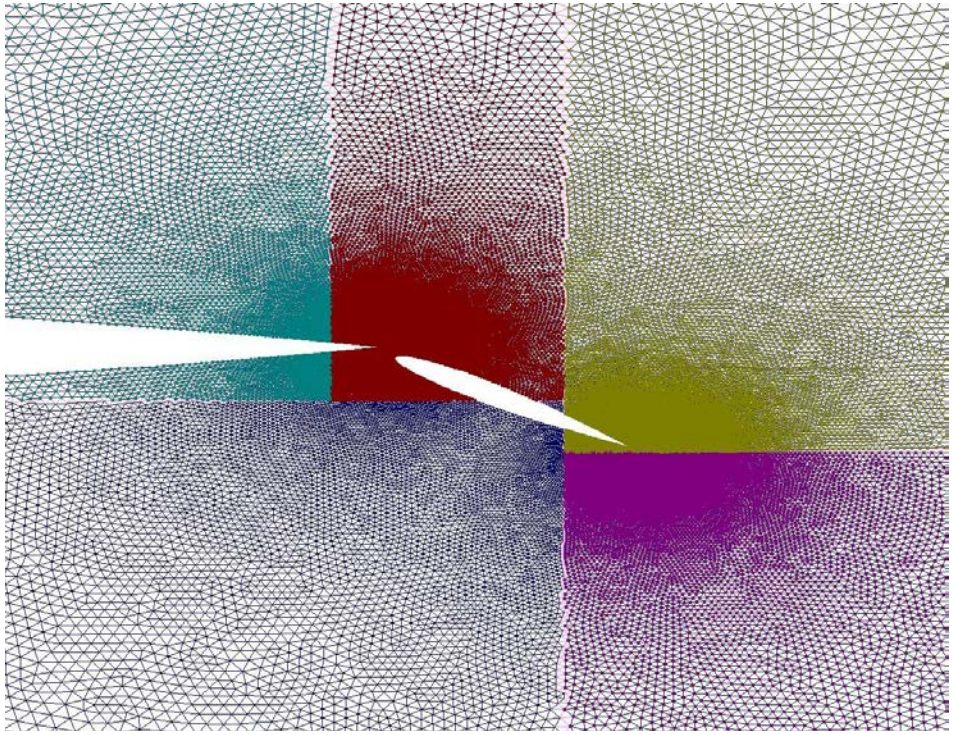


результат перераспределения малых блоков вершин

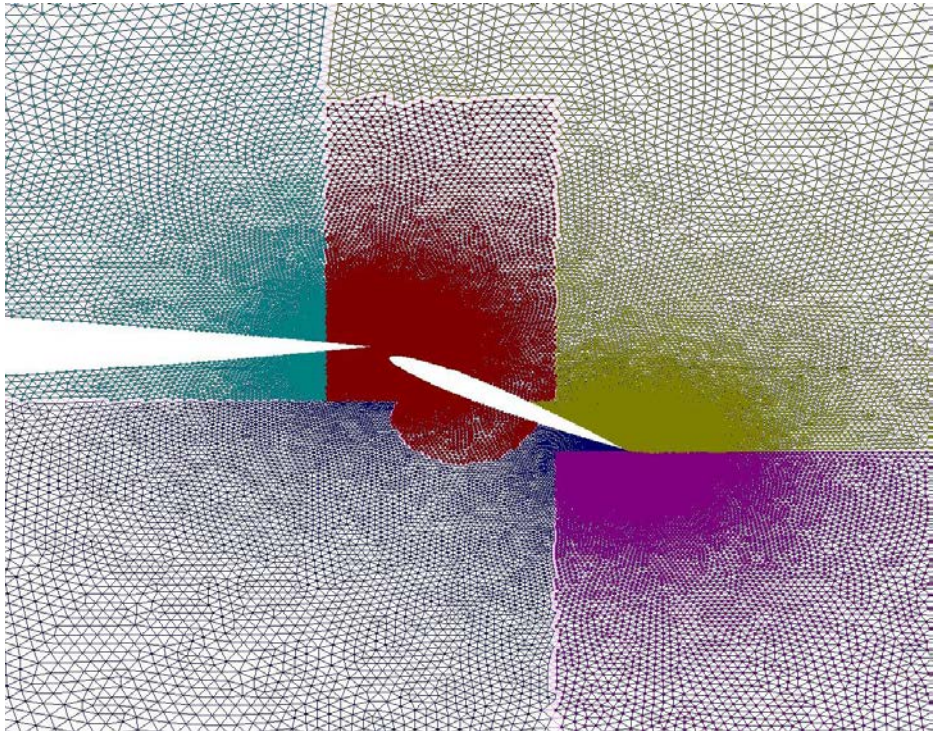


Фрагмент треугольной сетки из 75790 вершин

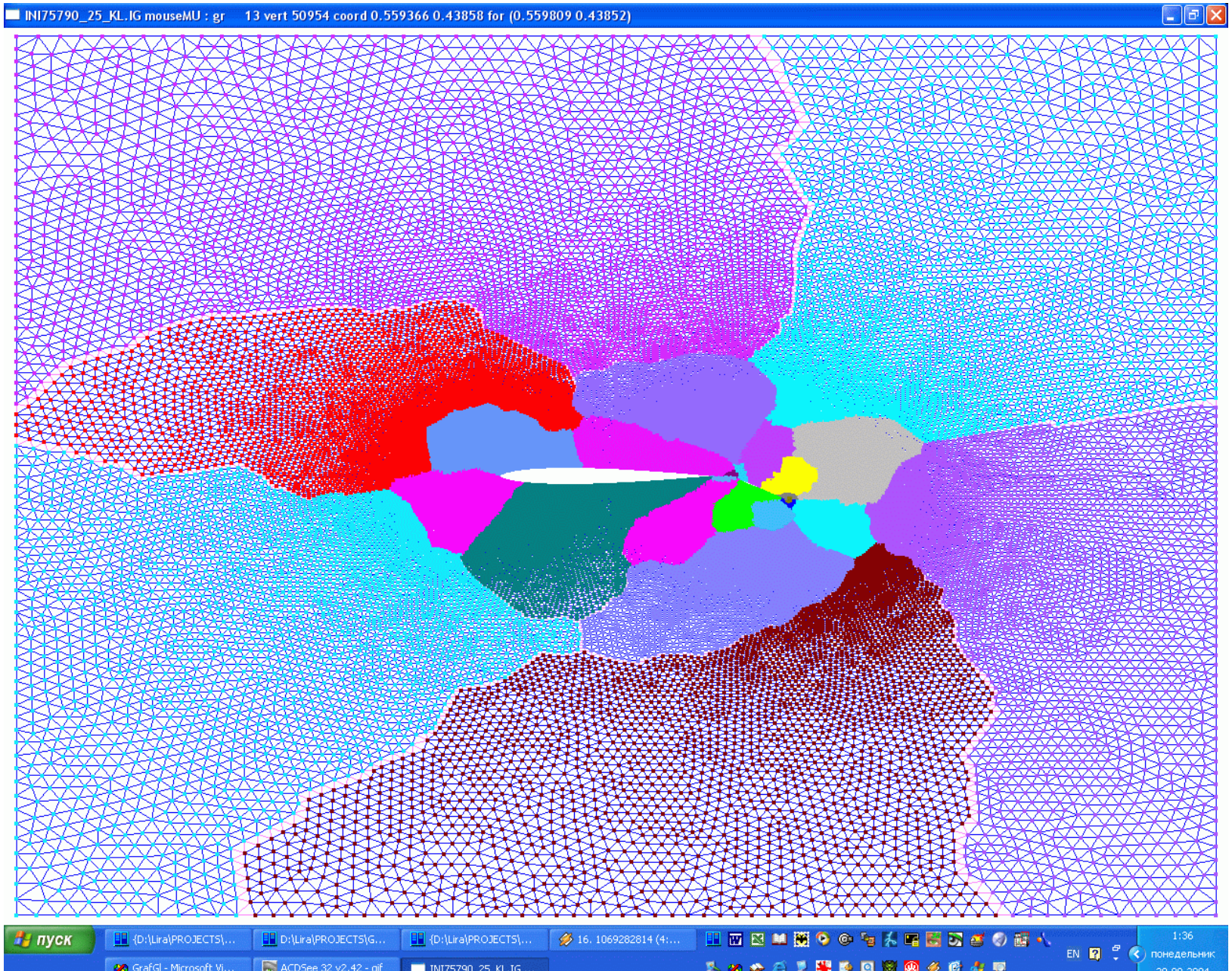
результат геометрической декомпозиции



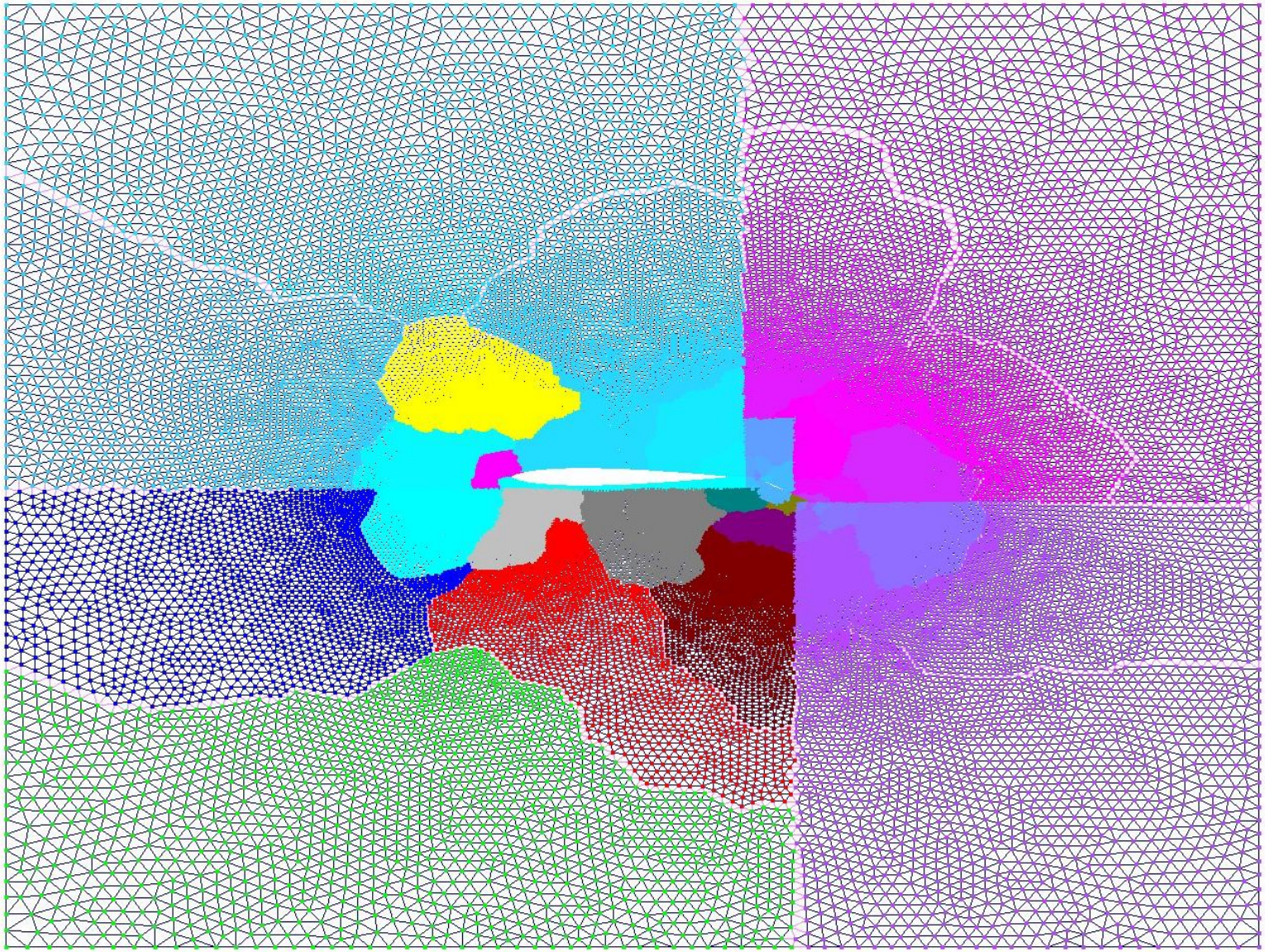
результат перераспределения малых блоков вершин



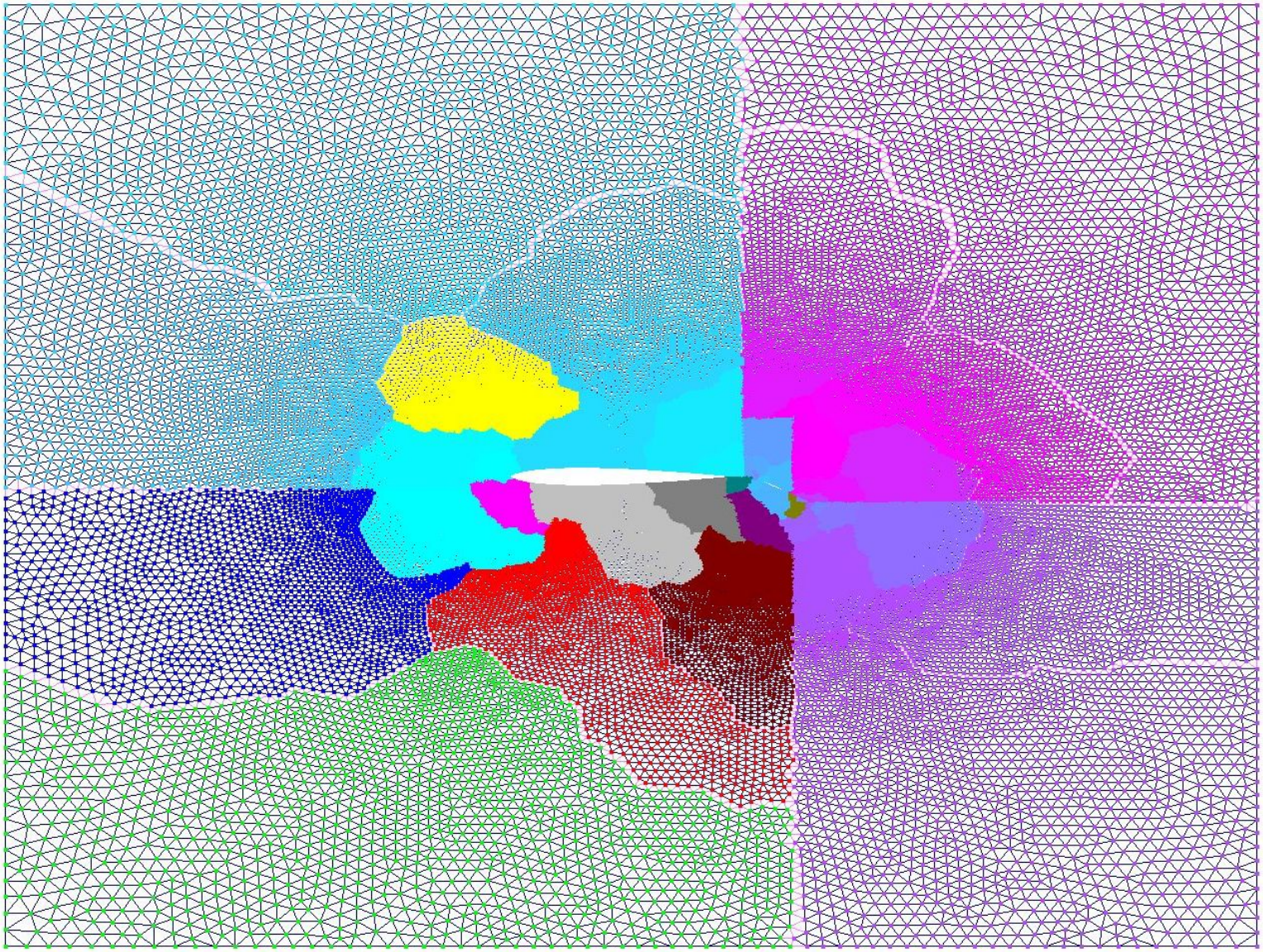
Инкрементный алгоритм, Dm=25



Результат локального разбиения сетки из 75790 вершин на 50 доменов на 5 процессорах



Результат сбора плохих групп доменов и их повторного разбиения

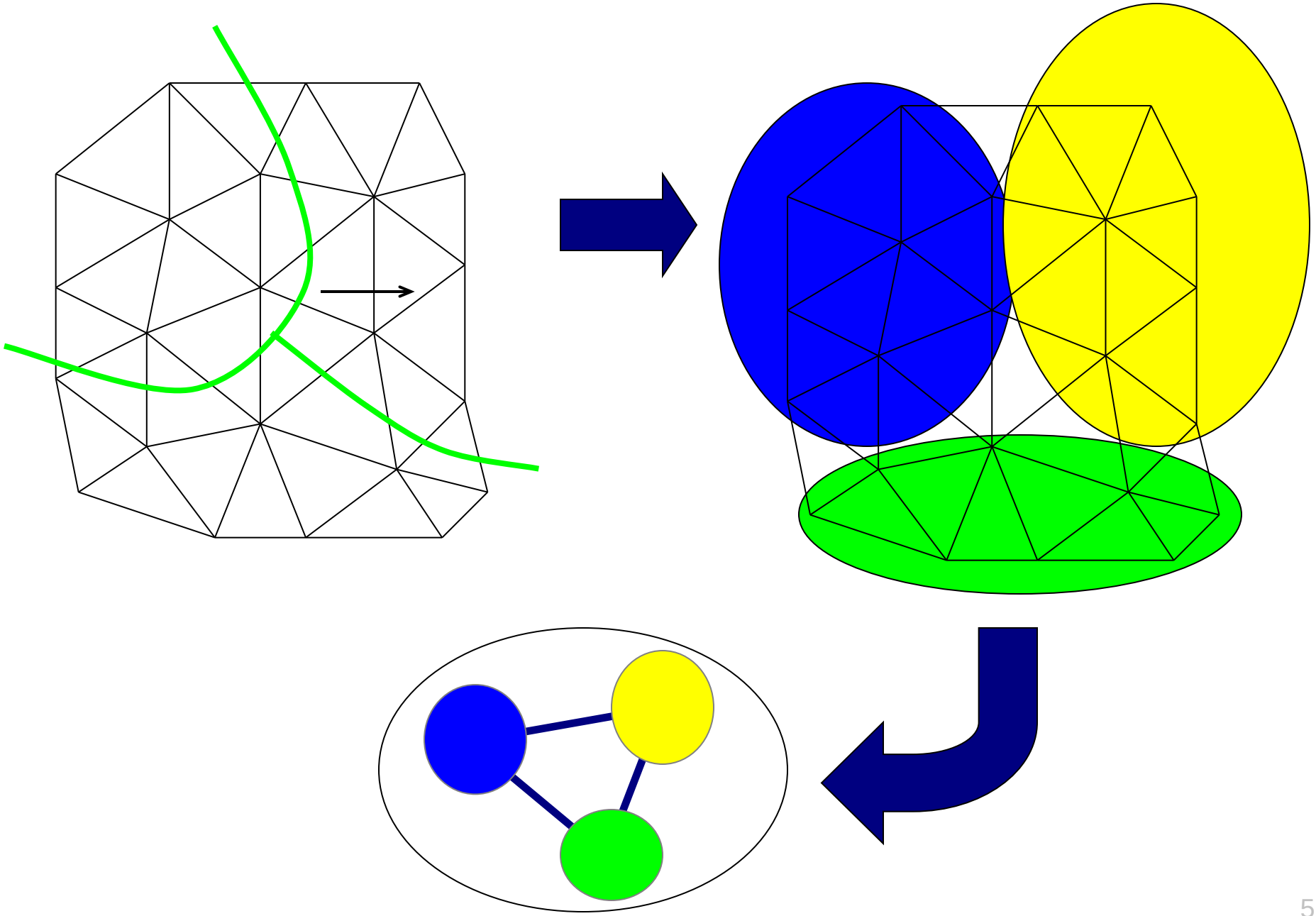


Разбиение тетраэдральной сетки, содержащей $2 \cdot 10^8$ узлов, на 125 процессорах

- вычисления производились на кластере СКИФ МГу (1250 4-хядерных процессоров, 60 TFlop/s)

		геометрическая декомпозиция		ParMETIS	
число доменов		80 000		20 000	
время		21 сек.		10 сек.	
число вершин в домене		2612	2613	2 328	10 932
мин.	макс.				
среднее число связей с соседними доменами		14		14	
число некомпактных доменов		229		364	

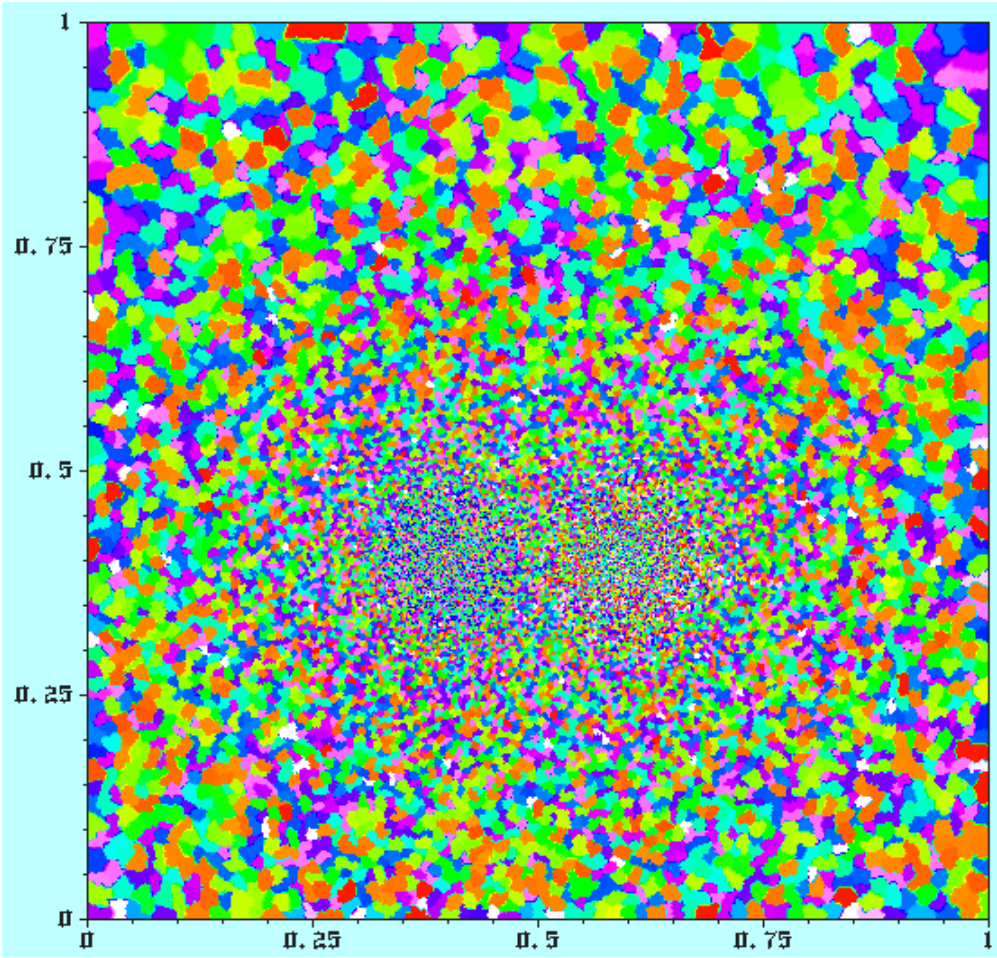
Формирование макрографа



Сетка микродоменов

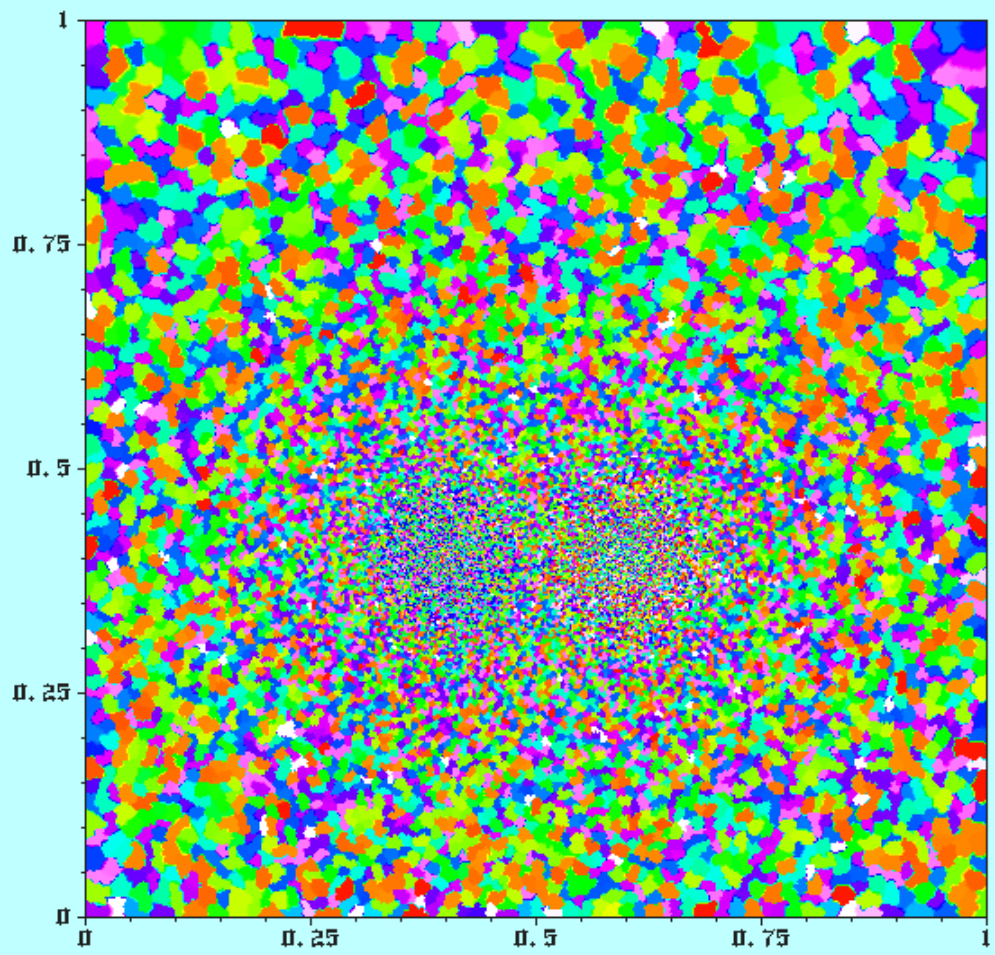
51 538 микродомен

в каждом около 20 узлов



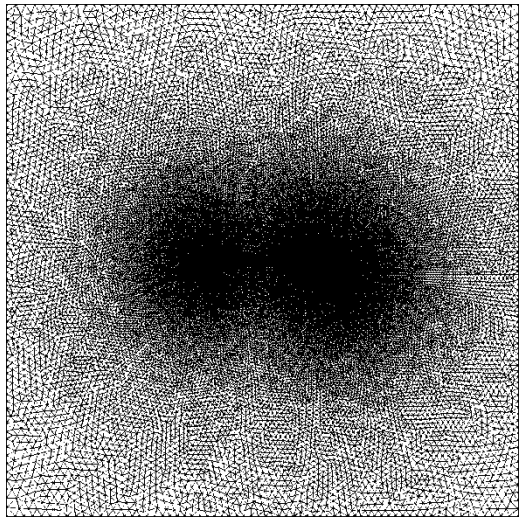
Сетка микродоменов

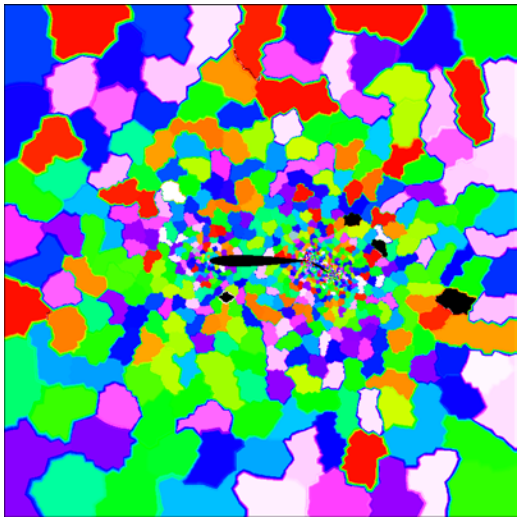
вес	число	% отн. число
12	3	0.01%
13	3	0.01%
14	15	0.03%
15	33	0.06%
16	228	0.44%
17	1 373	2.66%
18	5 433	10.54%
19	11 713	22.73%
20	14 218	27.59%
21	11 069	21.48%
22	5 737	11.13%
23	1 505	2.92%
24	192	0.37%
25	13	0.03%
26	2	0.00%
27	1	0.00%

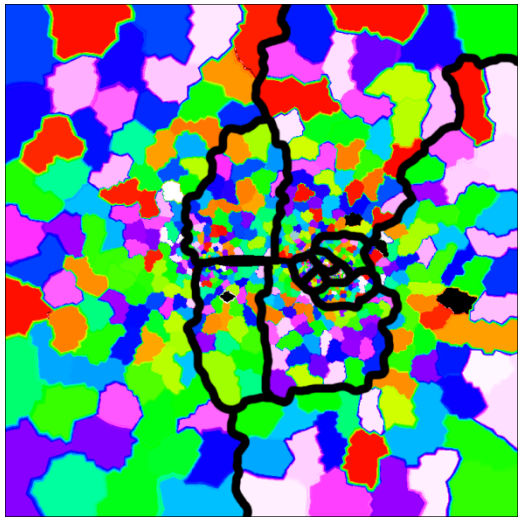


51 538 микродомен

Двухуровневое разбиение







I

II

Сетка предварительно разбивается на большое число *микродоменов*, образующих *макрограф*

Вершины *макрографа* распределяются по процессорам

Метод эффективен для сверхбольших сеток

Многоуровневое разбиение
Online декомпозиция
Динамическая балансировка

Недостаточно выполнять декомпозицию автономно

Препроцессинг

- **разбиение на микродомены**

Начальная декомпозиция

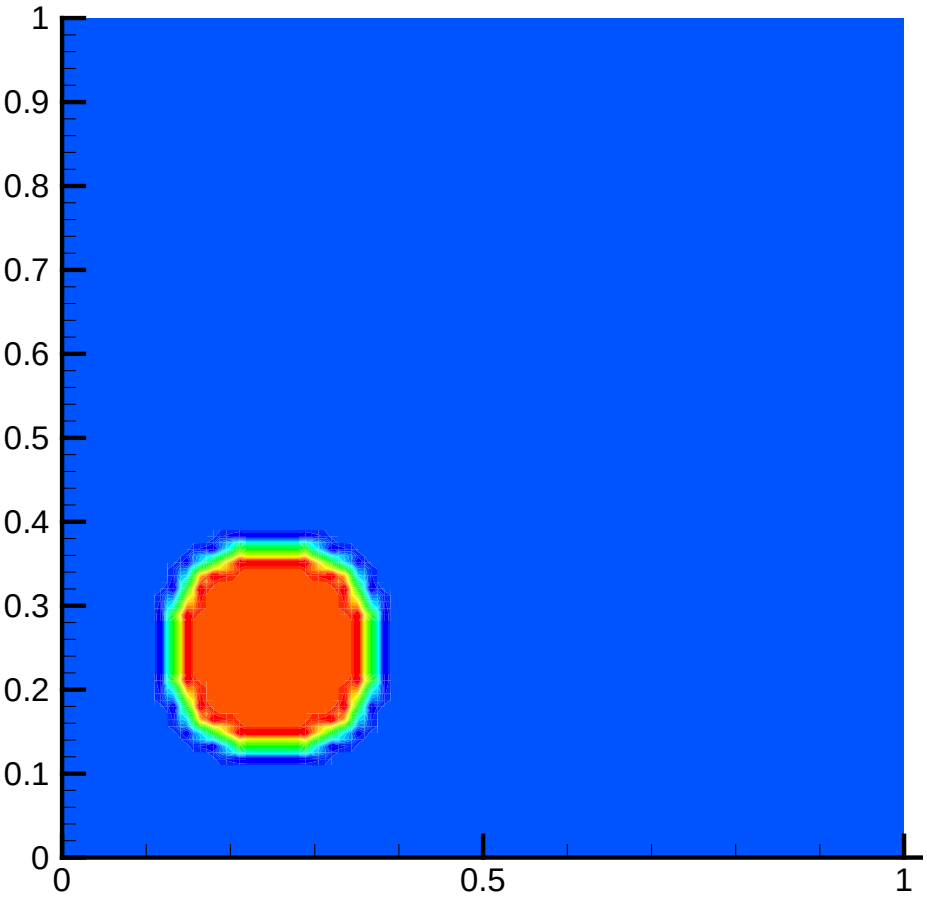
- **определение производительности ядер**
- **определение топологии системы**
- **определение характеристик интерконнекта**
- **определение трудоёмкости обработки микродоменов на ядрах**

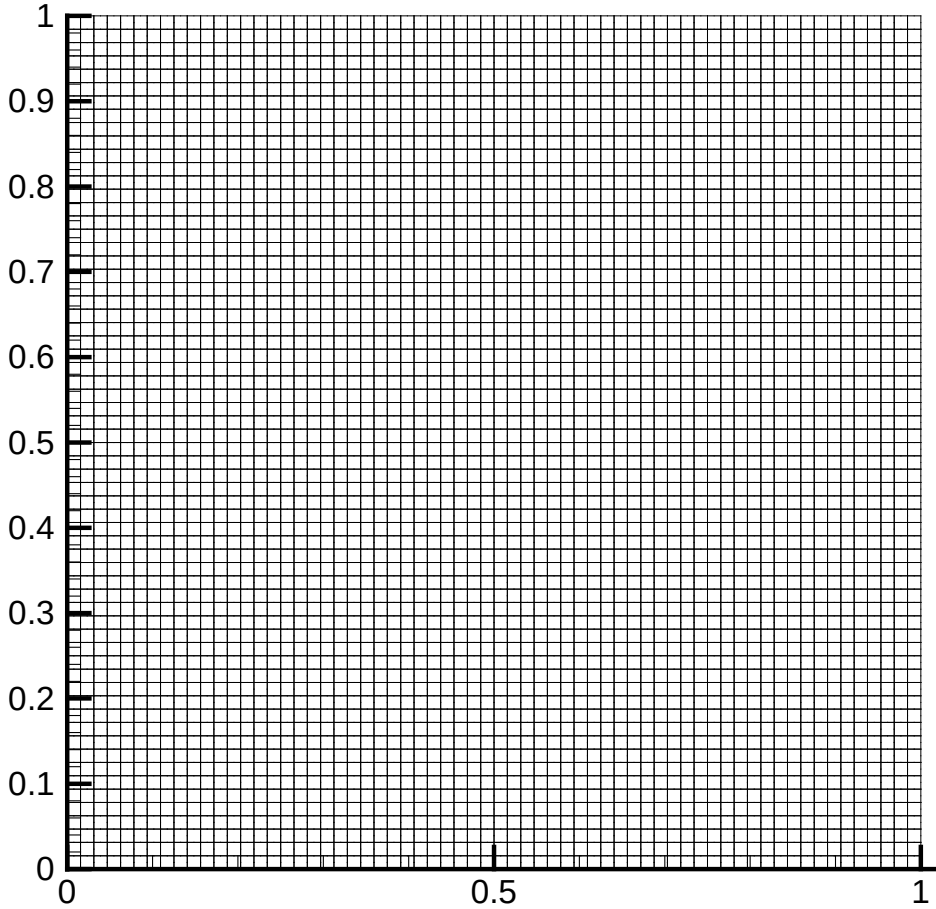
Динамическое перераспределение нагрузки процессоров

- **мониторинг производительности ядер**
- **перераспределения данных и вычислений**

Равномерная сетка

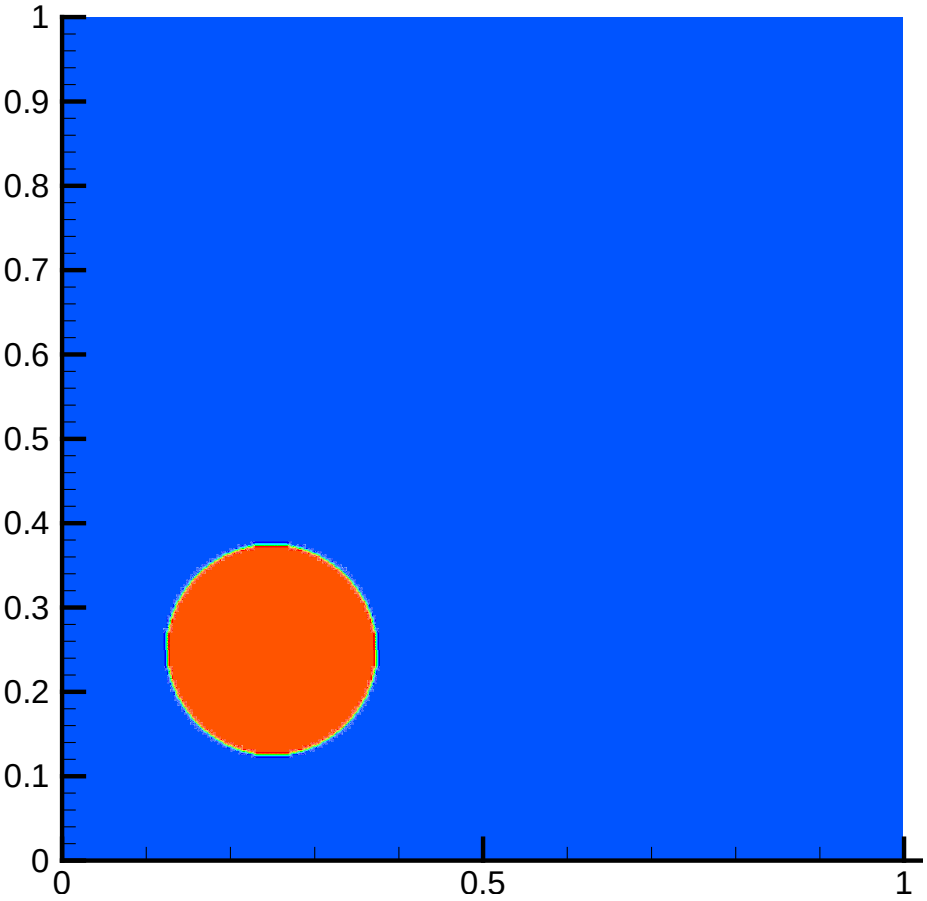
Слева – ??*круглое*?? пятно примеси

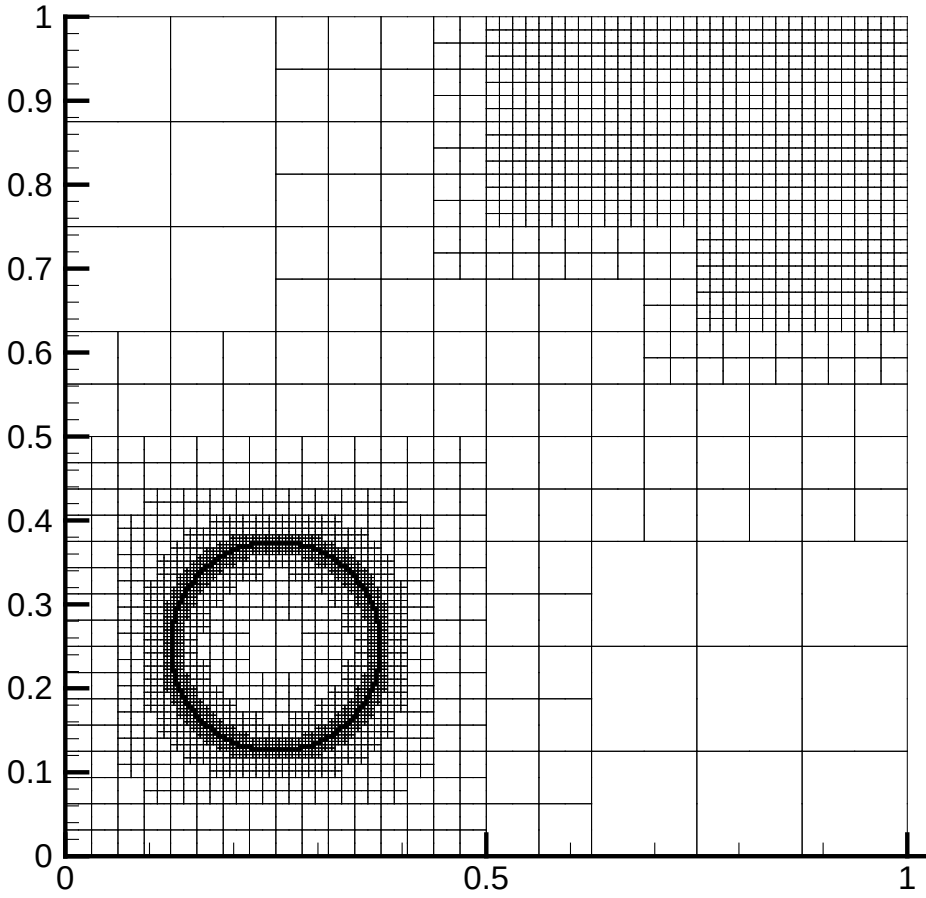




Адаптивная сетка

Слева – ***круглое*** пятно примеси

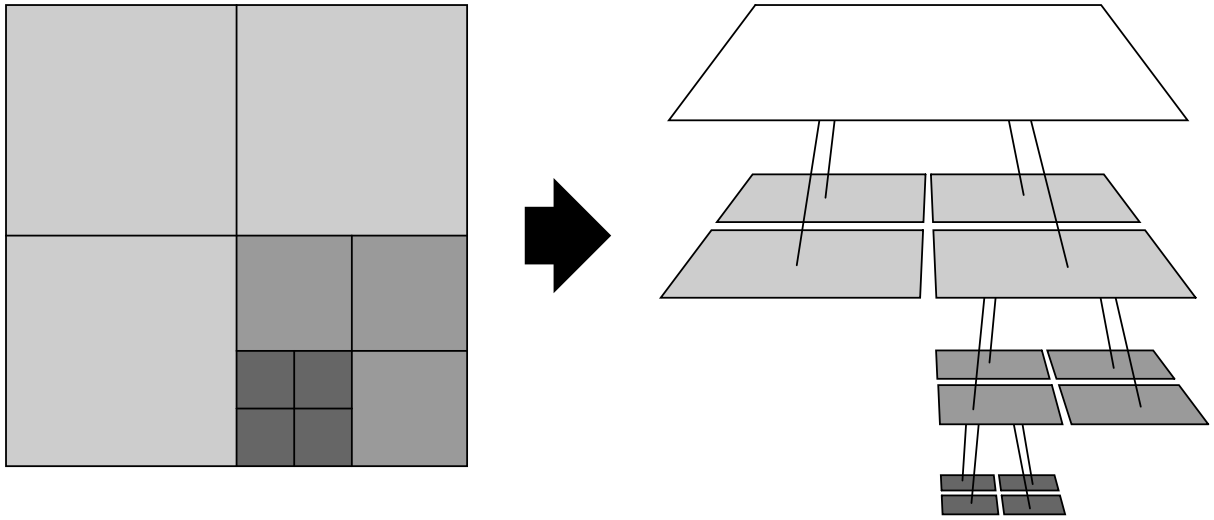




Адаптивные декартовы сетки

- Вначале сетка состоит из одной прямоугольной ячейки
- Каждая ячейка может быть разделена на четыре ячейки одинакового размера
- Если ячейки когда-то составляли одну ячейку, то они могут быть объединены обратно
- Каждая ячейка хранит величину, описывающую среднее значение неизвестной функции в пределах ячейки (метод конечных объёмов)

При данных предположениях сетку удобно хранить в виде четверичного дерева:



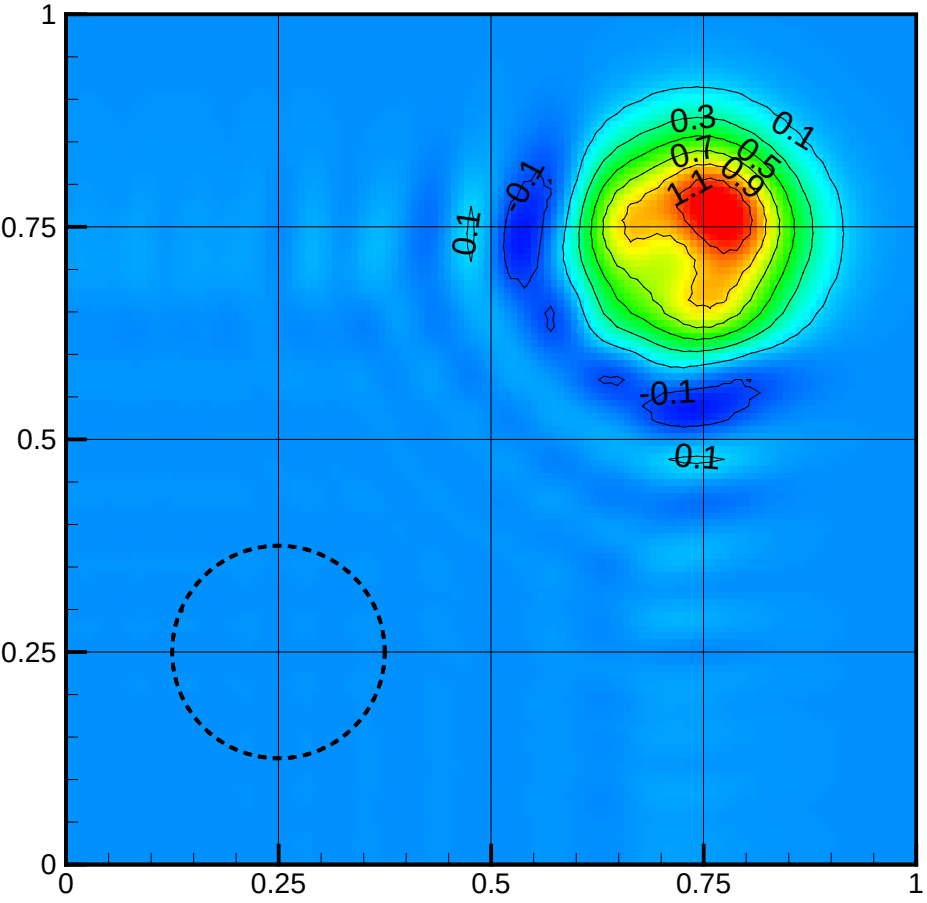
Дополнительные ограничения на размеры ячеек:

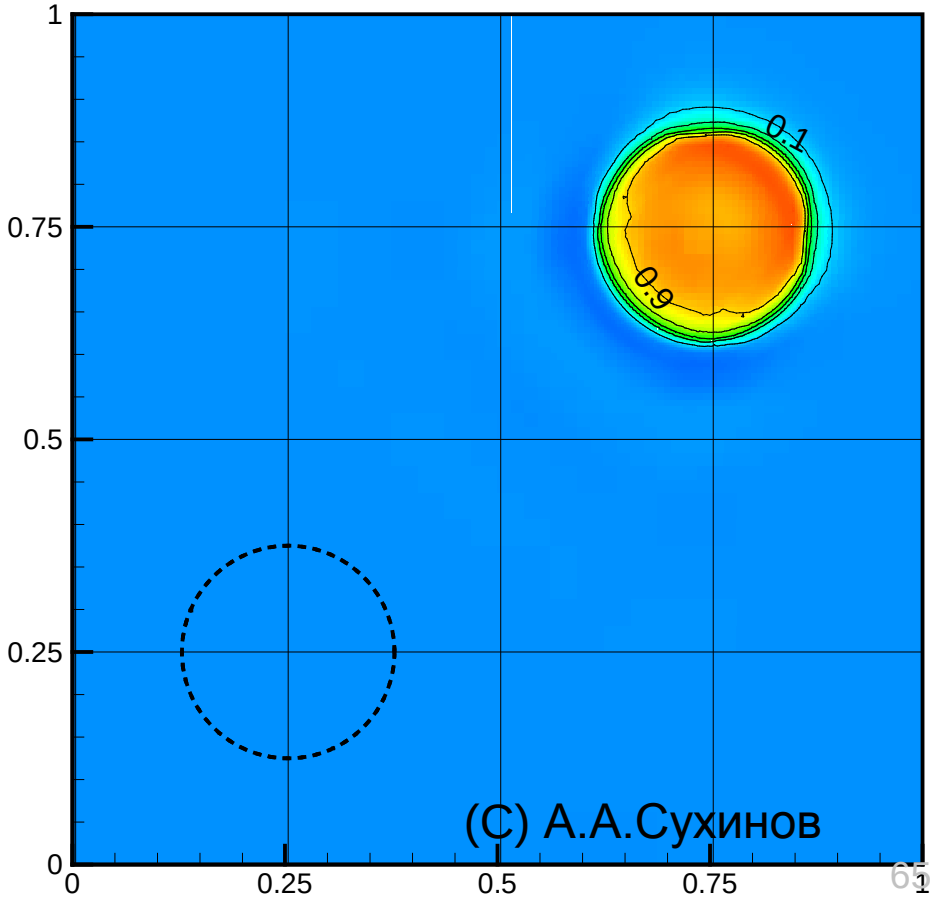
- Задан максимально допустимый размер ячеек
- Задан минимально допустимый размер ячеек
- Размеры соседних ячеек должны различаться не более, чем в 2 раза

(C) Сухинов А.А.

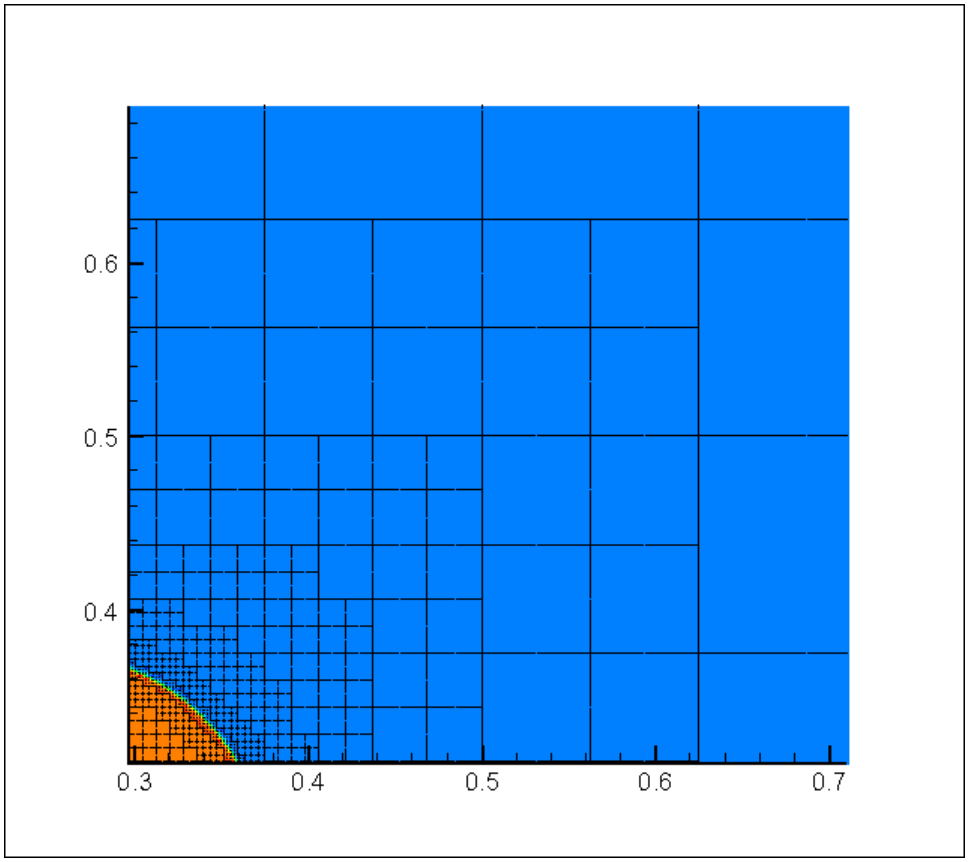
Сравнение с равномерной сеткой

На рисунках показаны результаты решения простейшей задачи переноса на равномерной (слева) и адаптивной (справа) сетках с одинаковым числом ячеек (4096 штук). Скорость переноса направлена под углом 45° к линиям сетки; начальное условие показано пунктиром

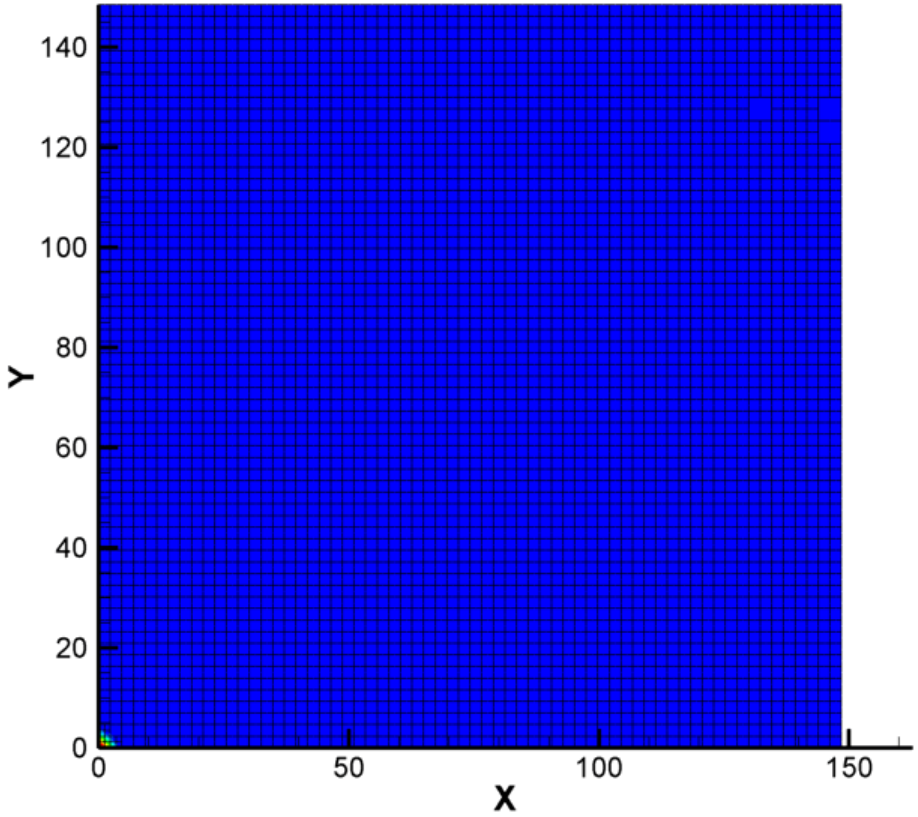


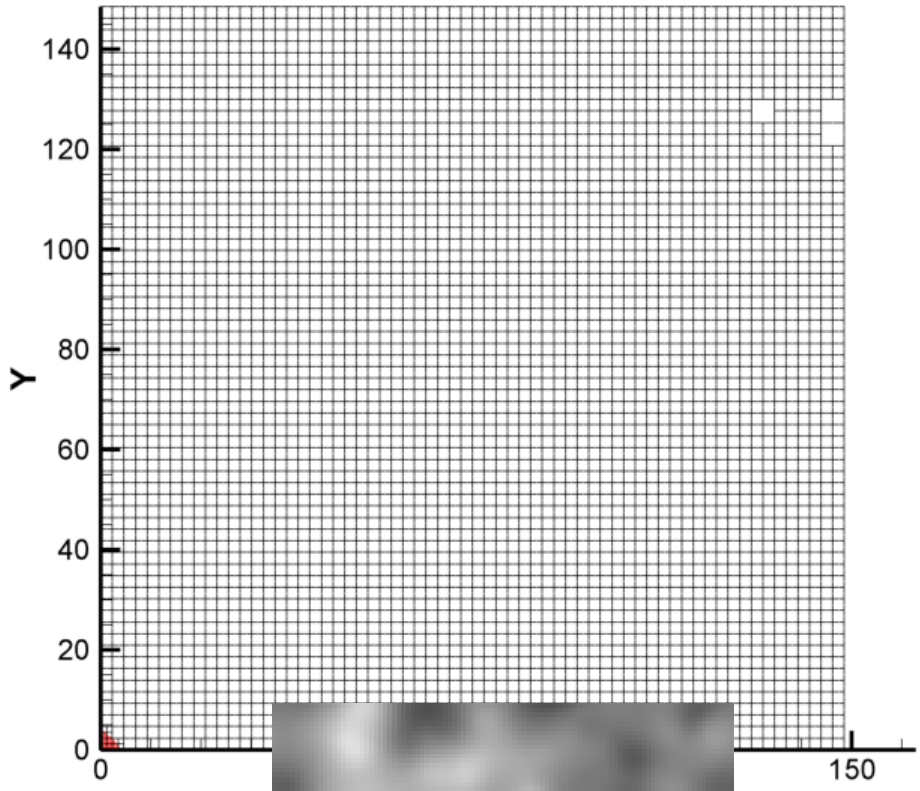


Адаптивная сетка

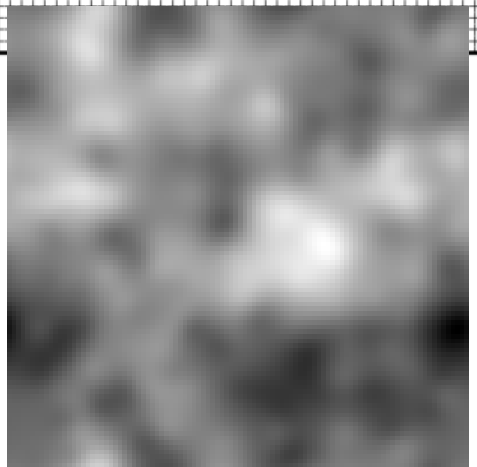


Масштабируемость алгоритмов и методов



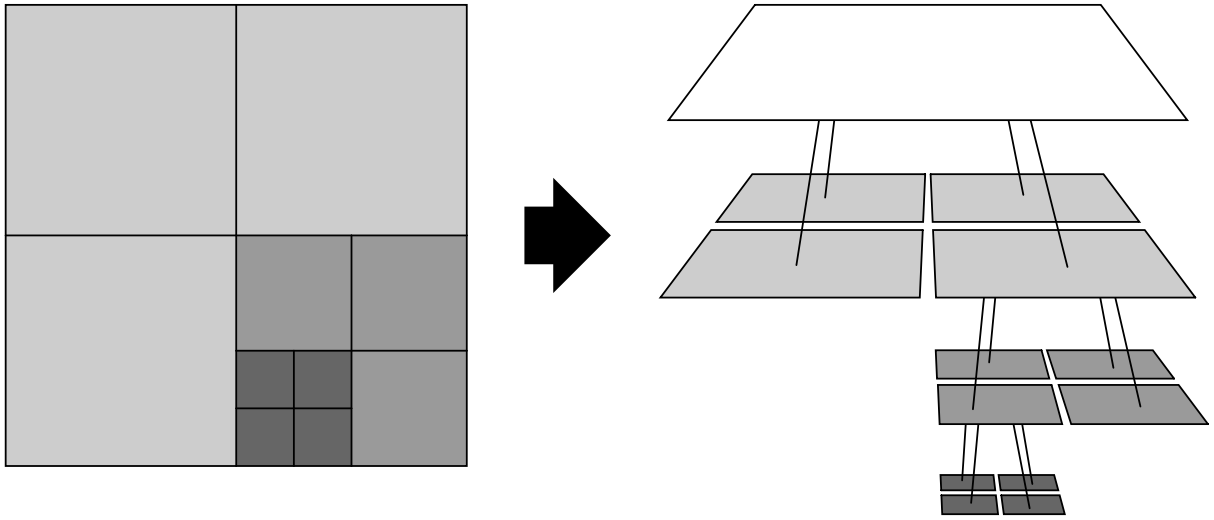


Заводнение месторождения.
Поле проницаемости с разбросом значений на **4** порядка

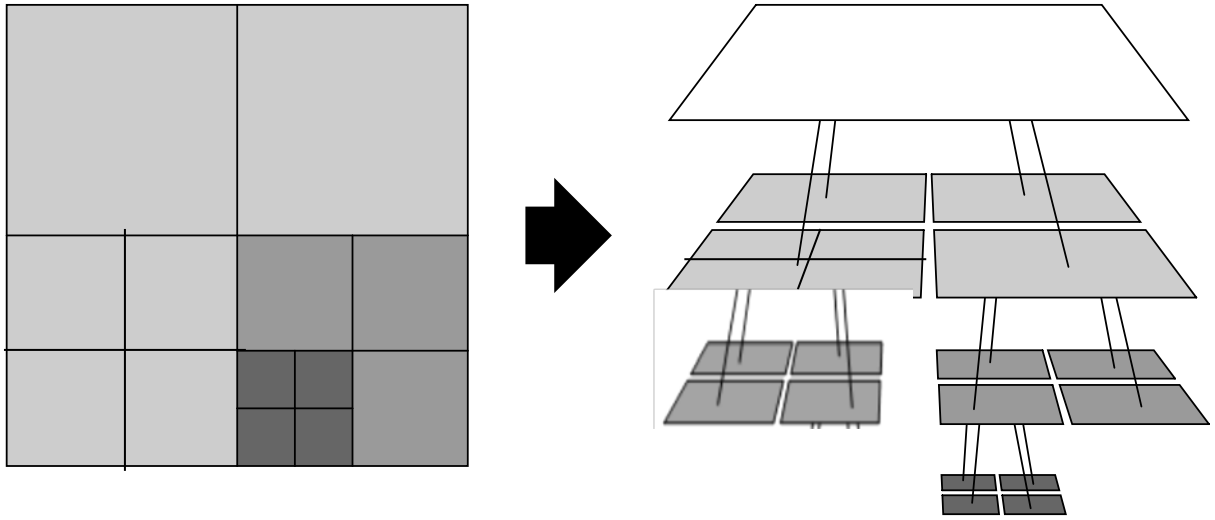


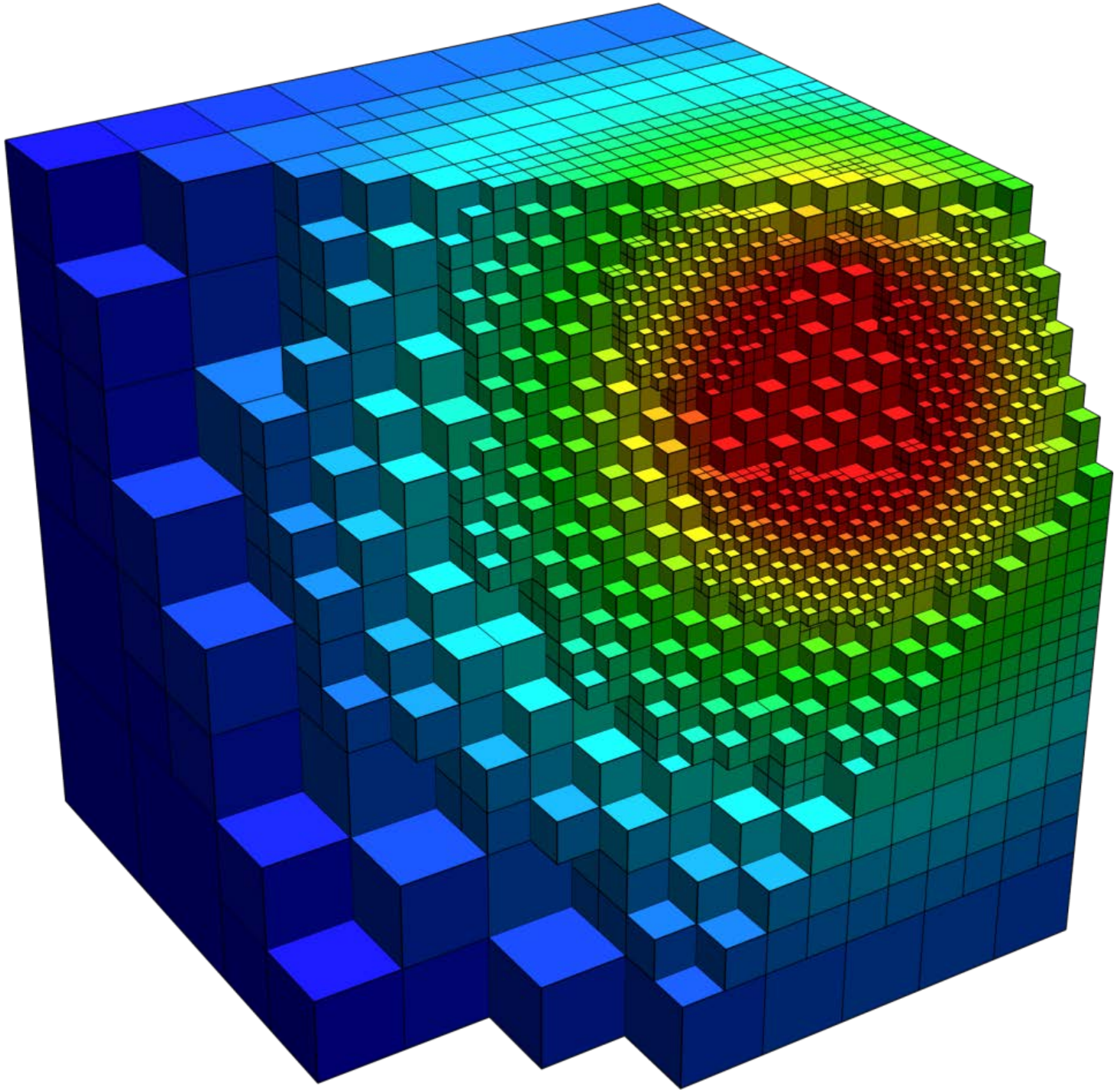
© А.А. Сухинов

Локальное измельчение



Структурный критерий





Этапы шага расчёта многокомпонентных течений

- Гидродинамика, явные схемы: локальные обмены
- Теплопроводность, обращение матриц: коллективные обмены
- Коллективные операции обмена
- Согласование термодинамических параметров – разброс времени вычислений (10 … 100 раз) в ячейках, содержащих разное число компонент нерегулярные обмены

Схема обменов соответствует методу декомпозиции по данным: обмен с ближайшими процессорами

- *взаимодействие соседних процессов для согласования величин на внутренних границах*

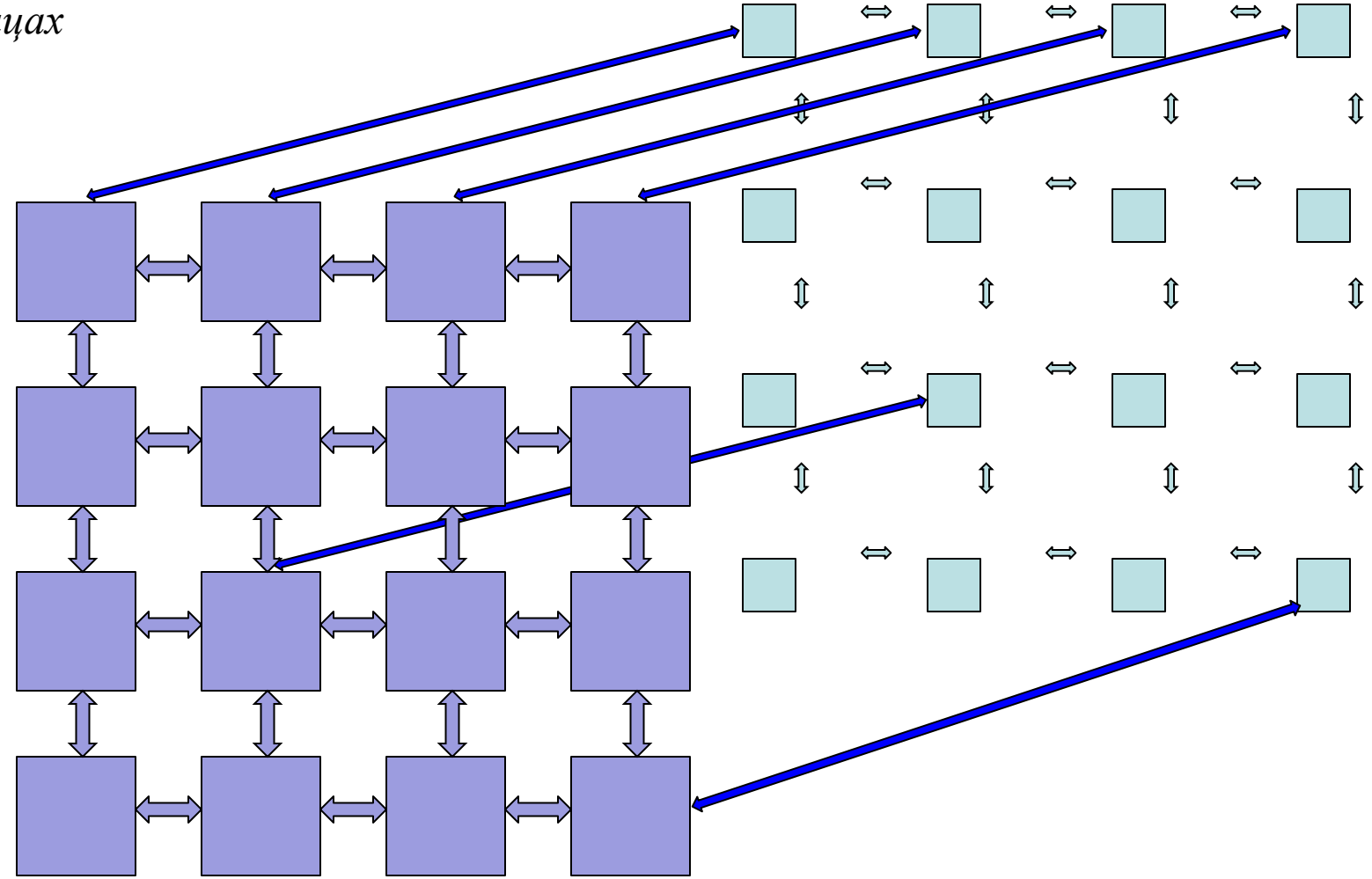
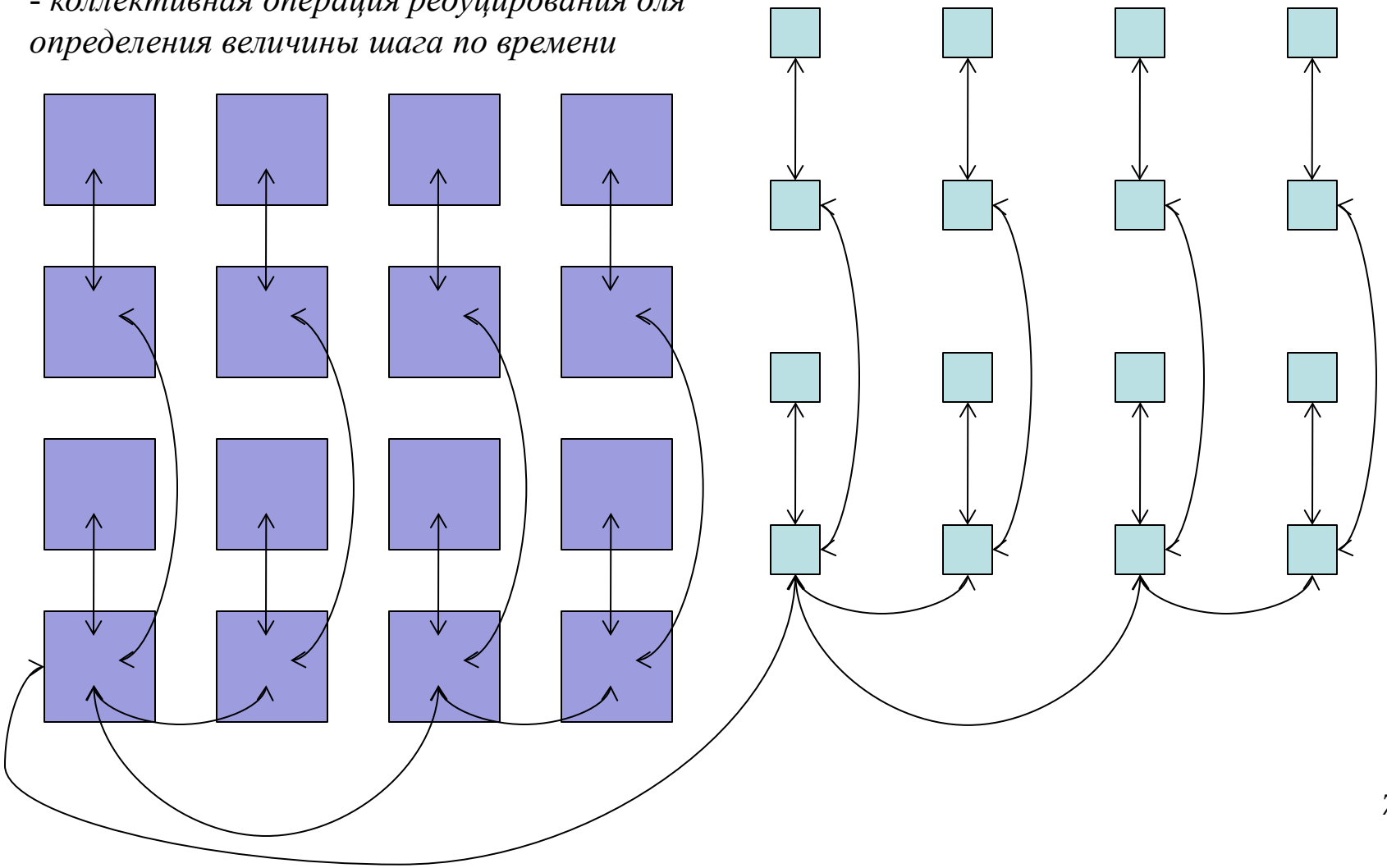


Схема обменов соответствует методу декомпозиции по данным: коллективная операция

- *коллективная операция редуцирования для определения величины шага по времени*



Этапы шага расчёта многокомпонентных течений

- Гидродинамика, явные схемы: локальные обмены
- Теплопроводность, обращение матриц: коллективные обмены
- Коллективные операции обмена
- Согласование термодинамических параметров – разброс времени вычислений (10 … 100 раз) в ячейках, содержащих разное число компонент нерегулярные обмены

Моделирование задач горения

$$\frac{\partial \mathbf{U}}{\partial t} + A\mathbf{U} = f, \qquad \mathbf{U} = \left(\rho, \rho y^{(i)}, \rho u, \rho v, E\right)^T \qquad f = (0, \omega_i, 0, 0, 0)^T$$

Здесь ρ - плотность,

$y^{(i)}$ – массовые доли i-х компонент,

u, v - скорости,

p - давление, E – полная энергия,

ω_I – скорости образования компонент.

I. Блок Газовой динамики:

$$\frac{\partial \mathbf{U}}{\partial t} + A\mathbf{U} = 0$$

$$\frac{\mathbf{U}^{j+1} - \mathbf{U}^{j}}{\Delta t} + \frac{1}{2}\left(A\mathbf{U}^{j+1} + A\mathbf{U}^{j}\right) = 0.$$

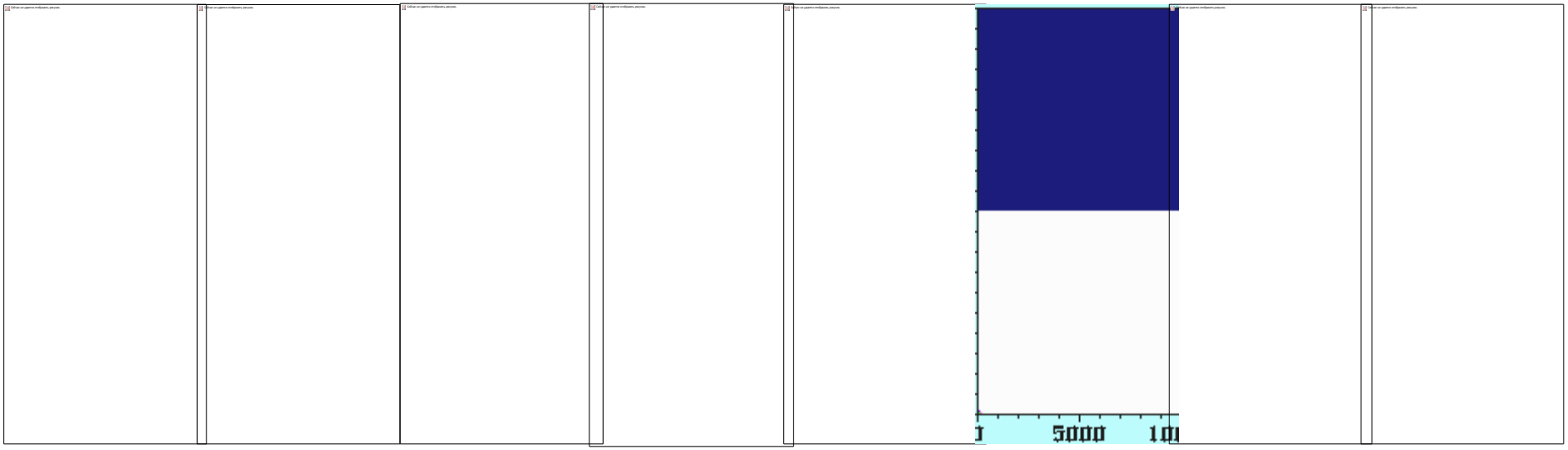
II. Блок химической кинетики:

$$\frac{d\mathbf{U}}{dt} = f, \qquad f = (0, \omega_i, 0, 0, 0)^T$$

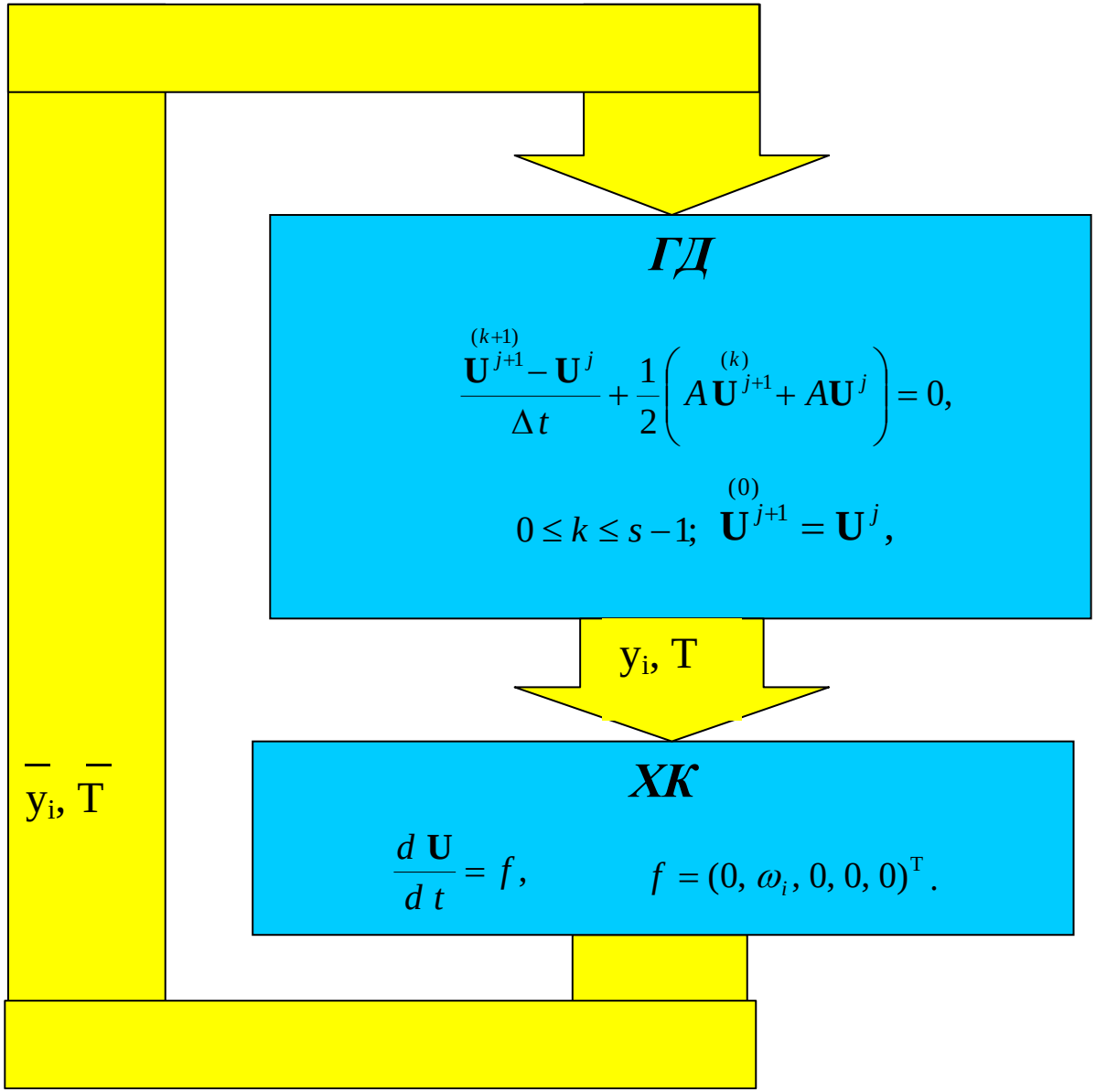


Methane combustion

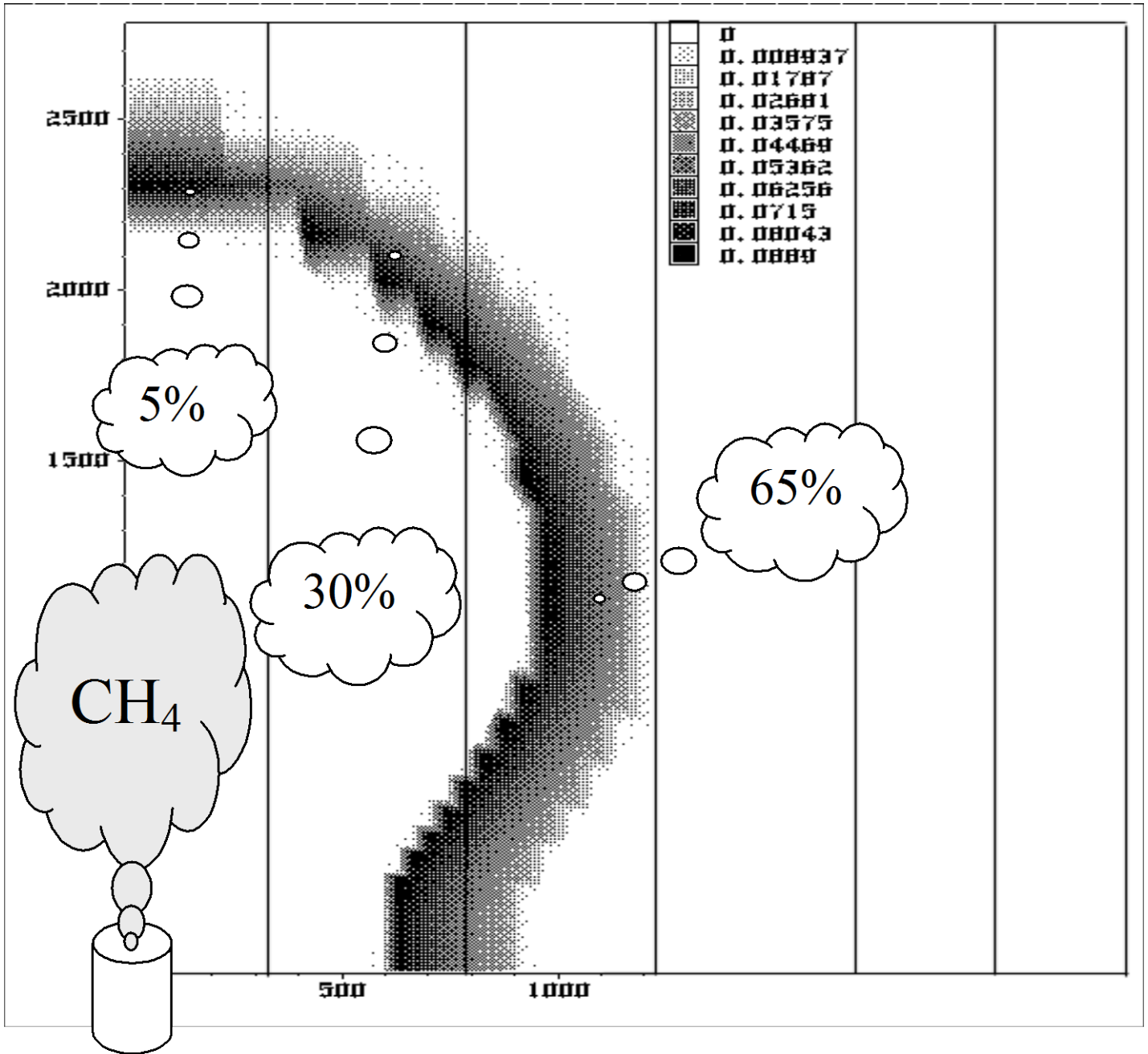
CH_4 NO NO_2 N_2O CH_3 CO CO_2 T



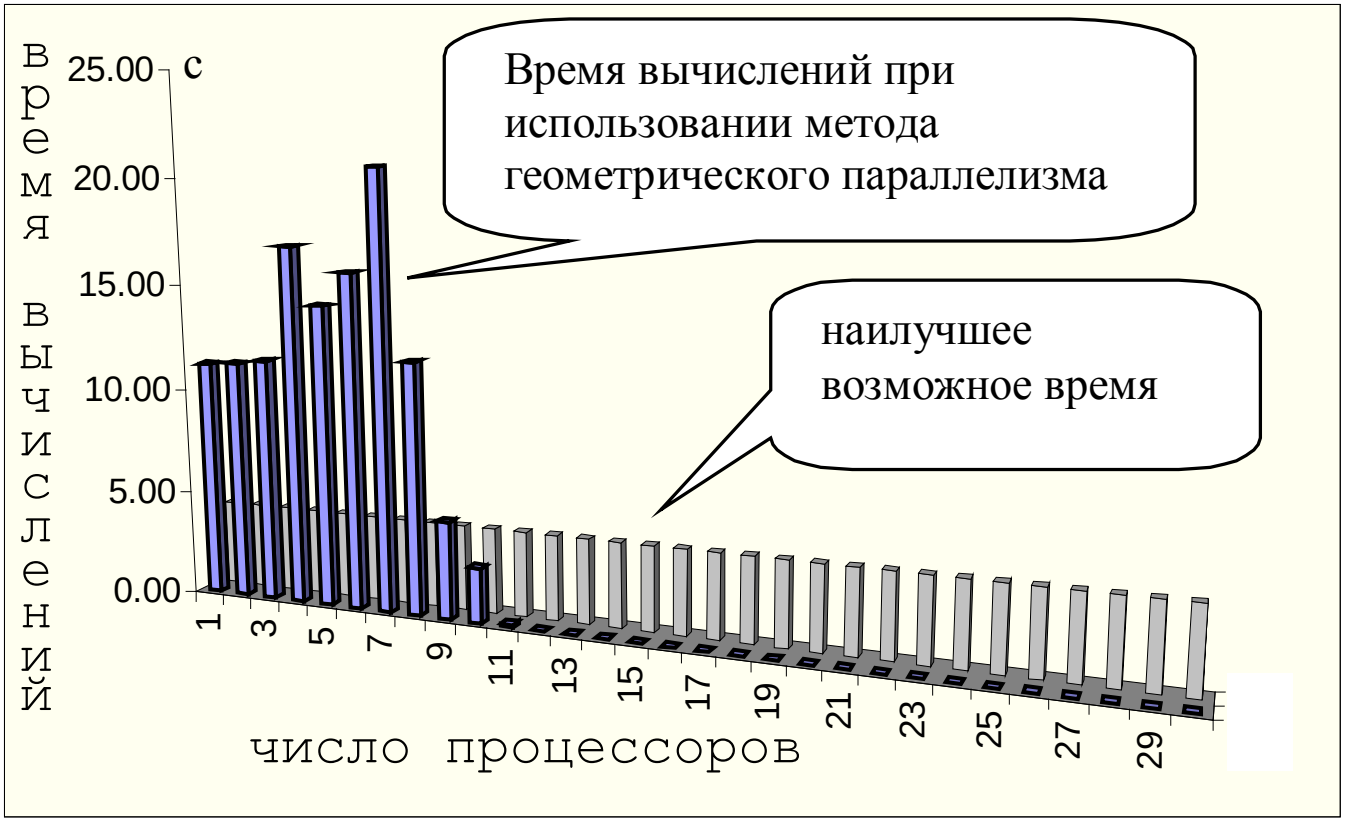
Решение методом суммарной аппроксимации



Метановый факел

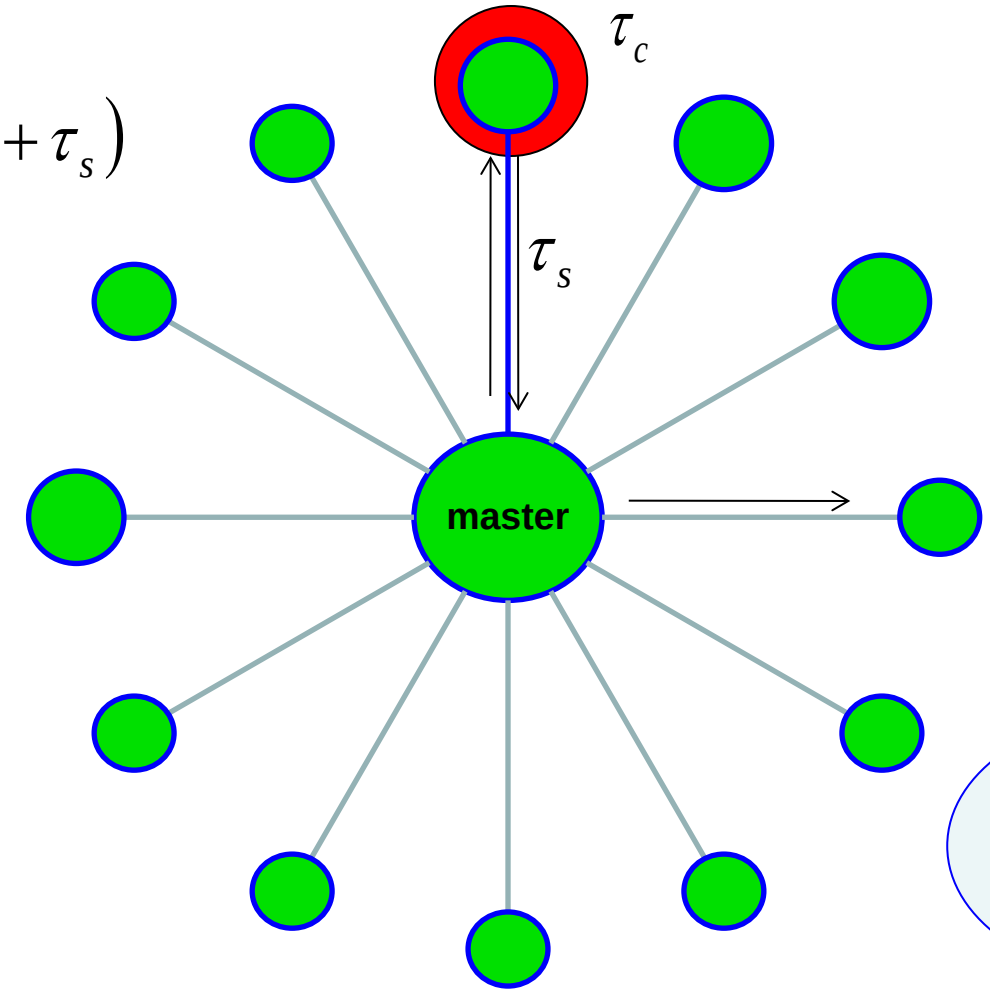


Распределение времени счета



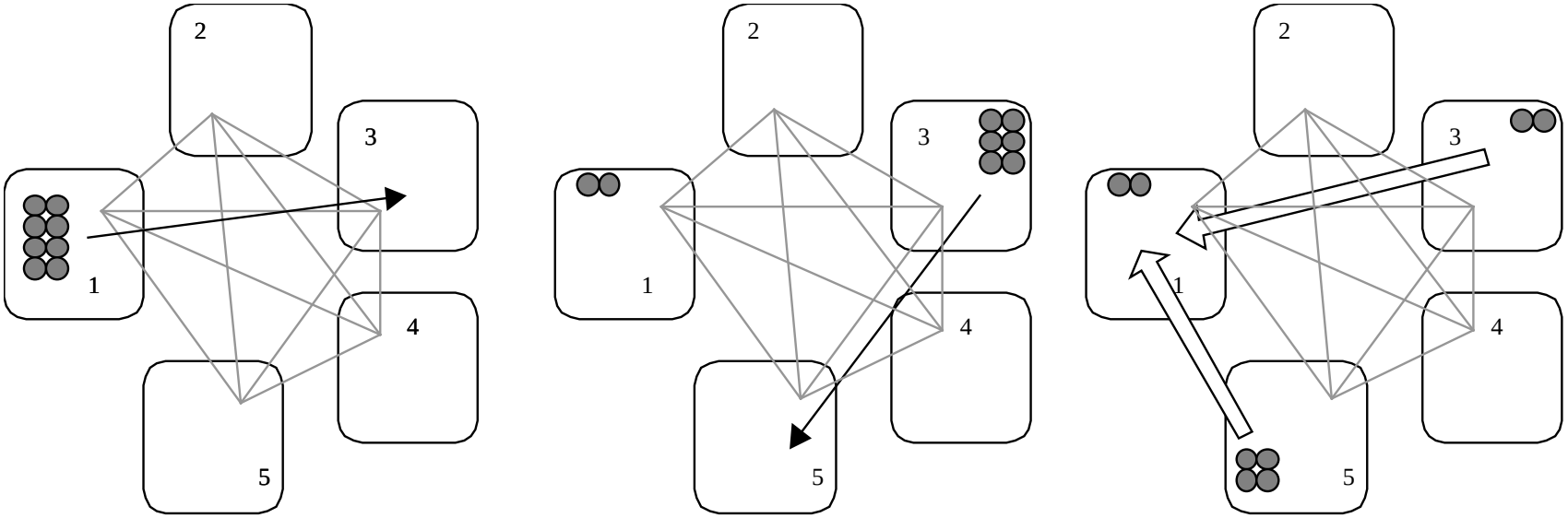
Метод коллективного решения

$$T_p = \frac{N}{p}(\tau_c + \tau_s)$$



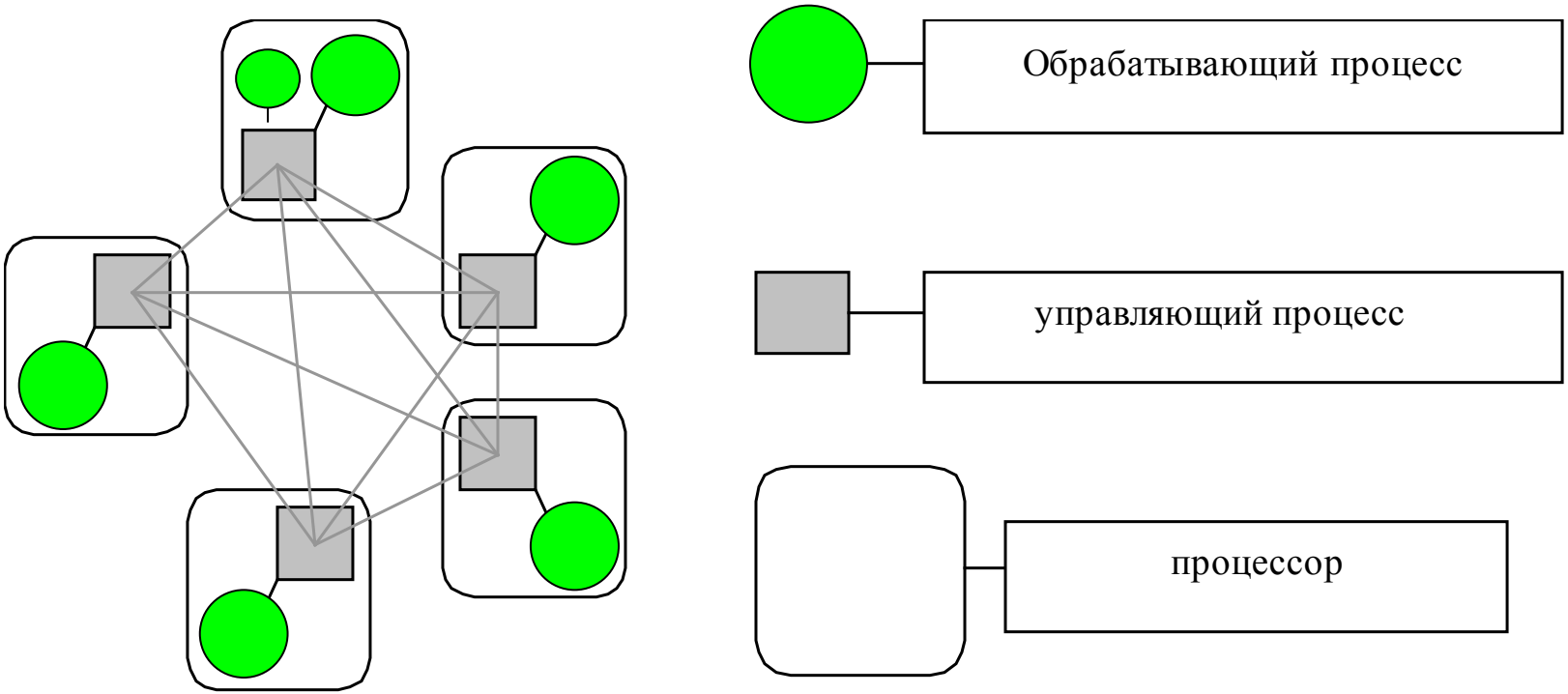
$$p_{\max} = \frac{\tau_c}{\tau_s}$$

Серверный параллелизм



Серверный параллелизм

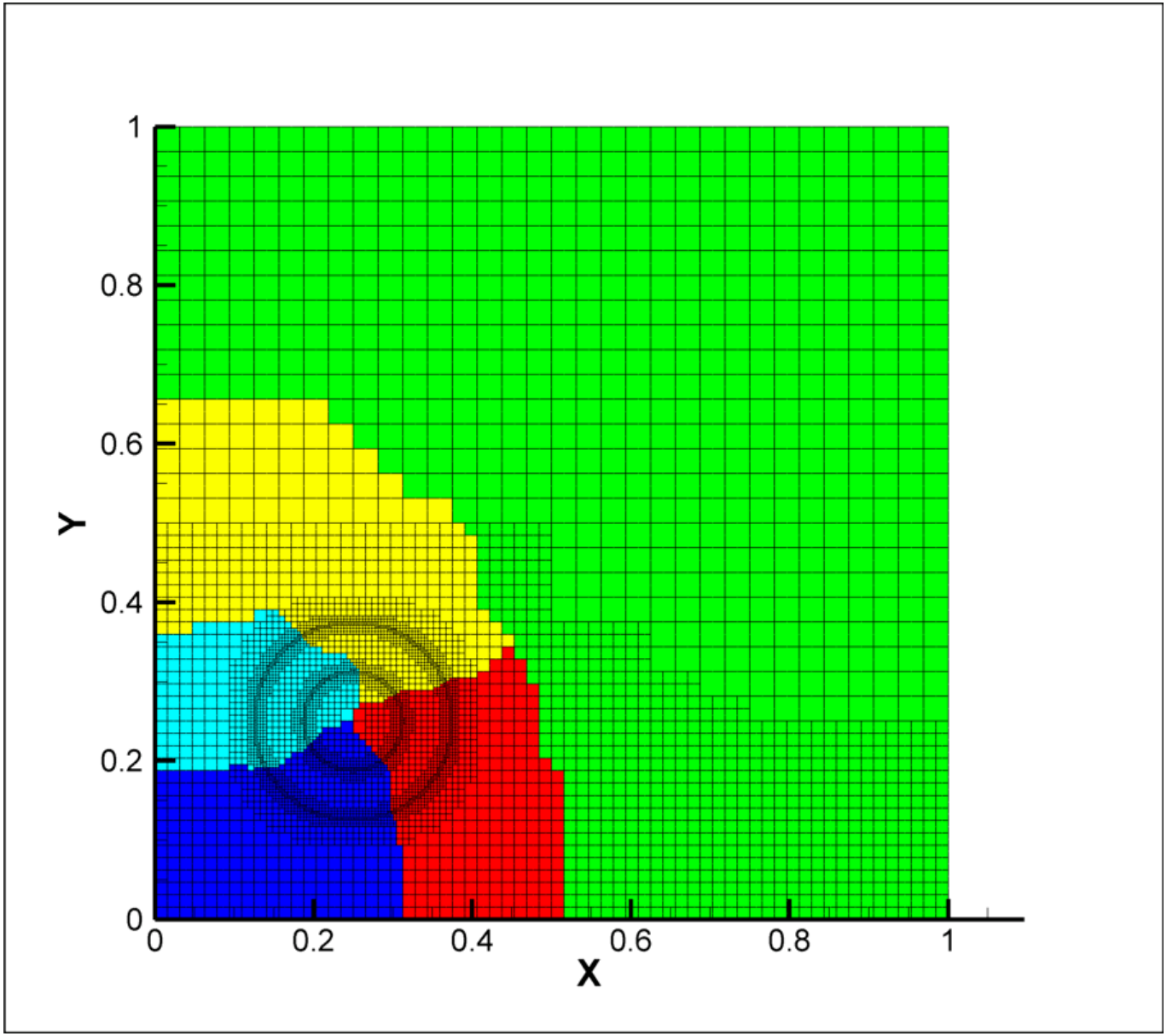
- Серверный параллелизм - метод динамической балансировки загрузки, применимый в случае обработки ***распределенного*** множества элементарных заданий ***непредсказуемой*** трудоёмкости

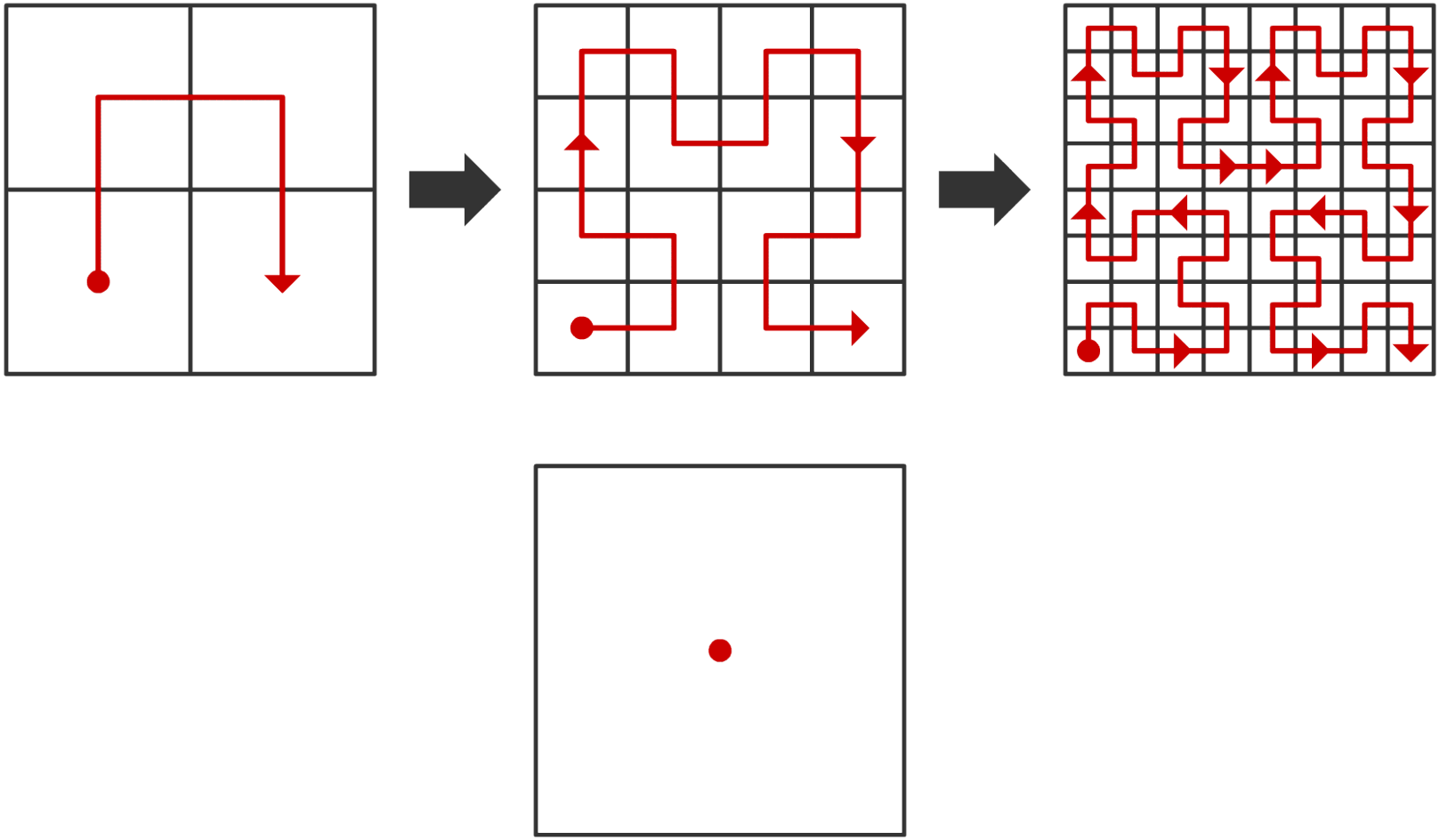


Локально измельчаемые сетки
Причины потери производительности

- Последовательный алгоритм
  - Дополнительные операции для согласования потоков на границах областей с разным уровнем измельчения
- Параллельный алгоритм
  - Статические сетки
    - Дисбаланс вычислительной нагрузки
    - Передача данных между процессорами на шагах расчёта
  - Динамическая адаптация
    - Меняющийся уровень вычислительной нагрузки на процессорах
    - Формирование новой сетки
    - Декомпозиция новой сетки
    - Перераспределение данных между процессорами

Декомпозиция пакетом Metis





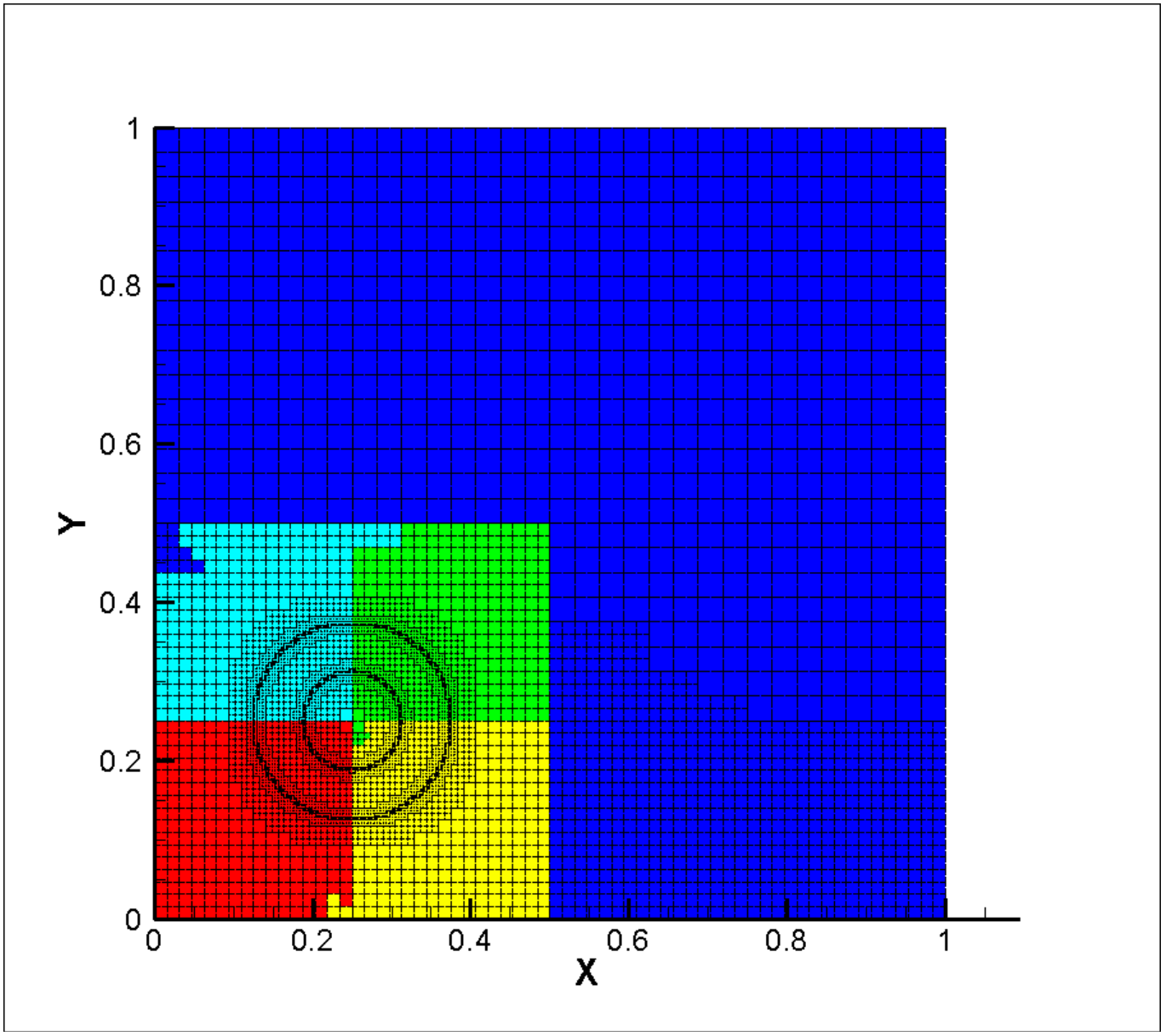
Hilbert-curve ordering

This ordering can be built by simple recursive procedure.

When mesh changes locally, Hilbert curve changes locally too.

It cannot be used for parallel computations due to chain dependence of elements.

Декомпозиция вдоль кривой Гильберта



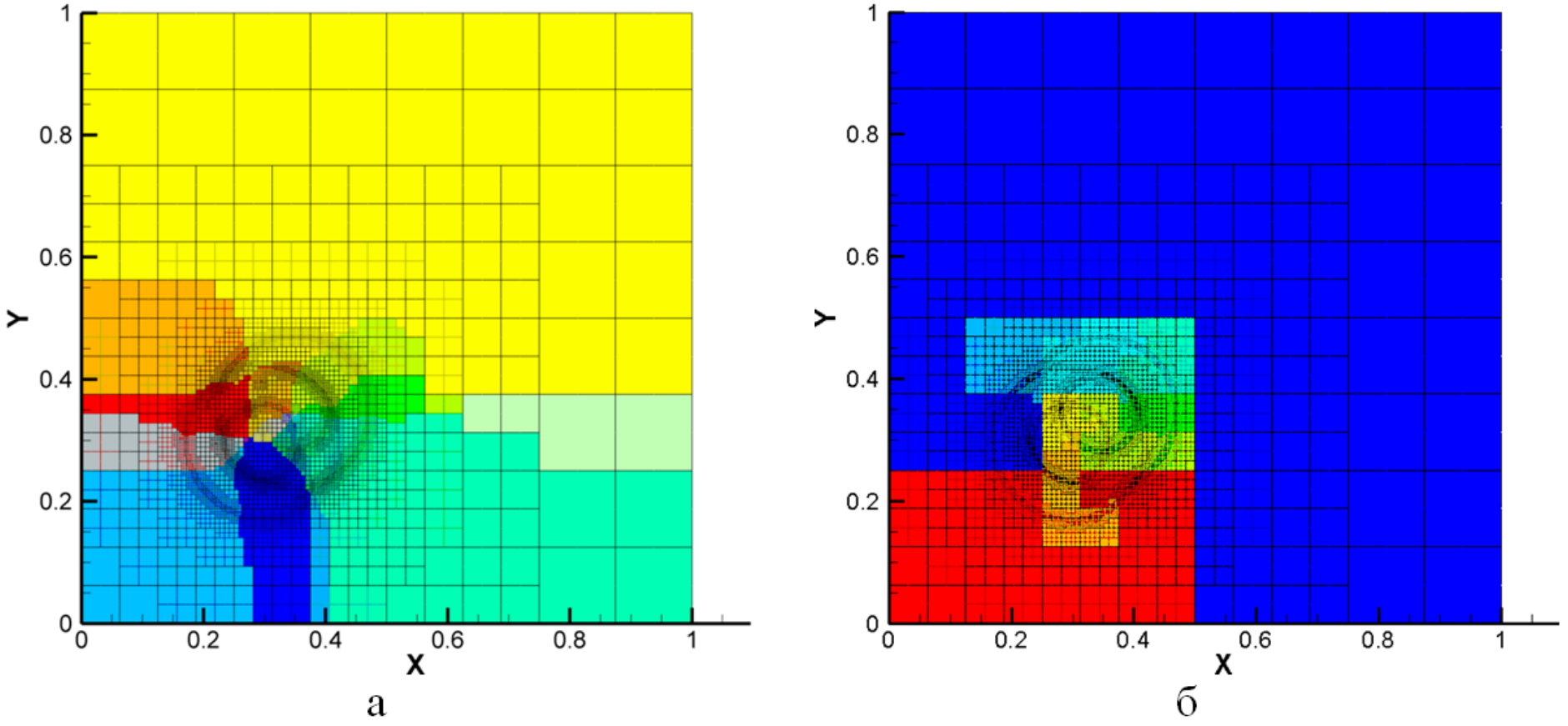
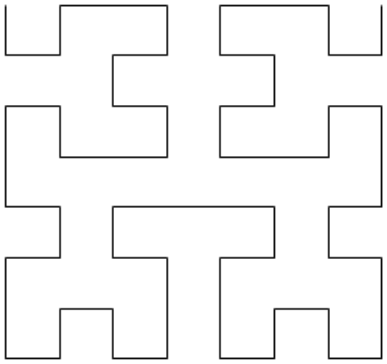
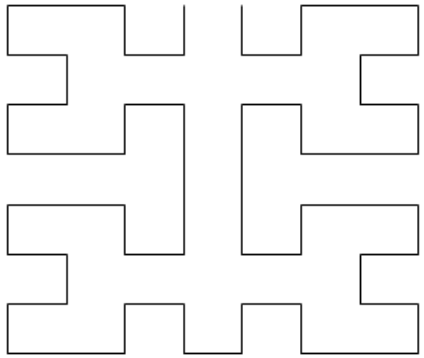


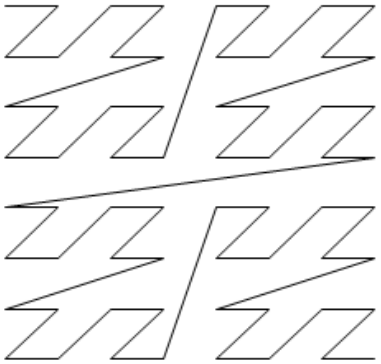
Рис. [14]. Пересечение доменов, полученных с помощью иерархического метода (а) и с помощью кривой Гильберта (б)



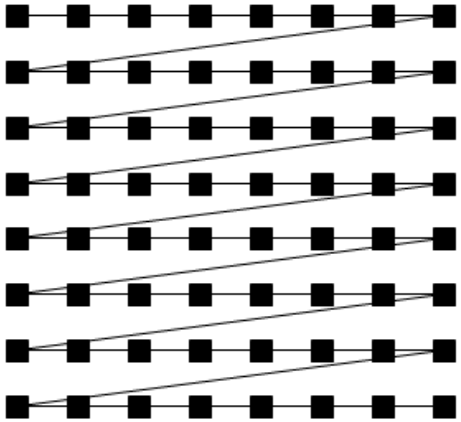
Кривая Гильберта



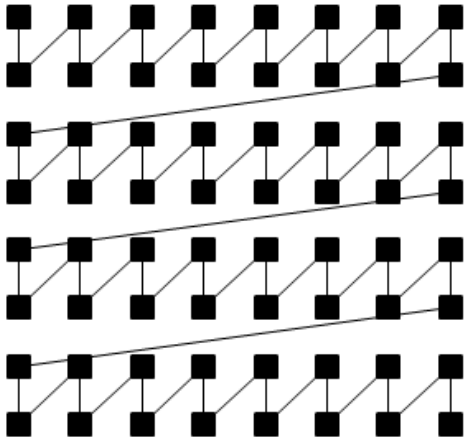
Кривая Мура



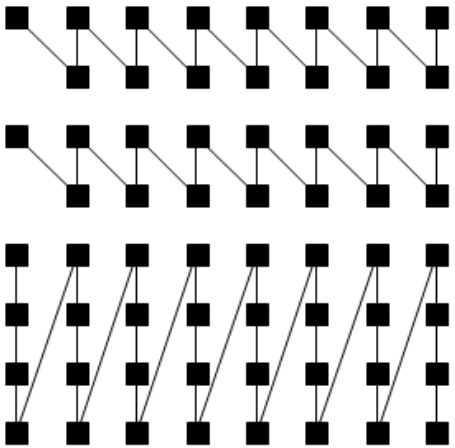
Кривая Мортона



Растровая кривая



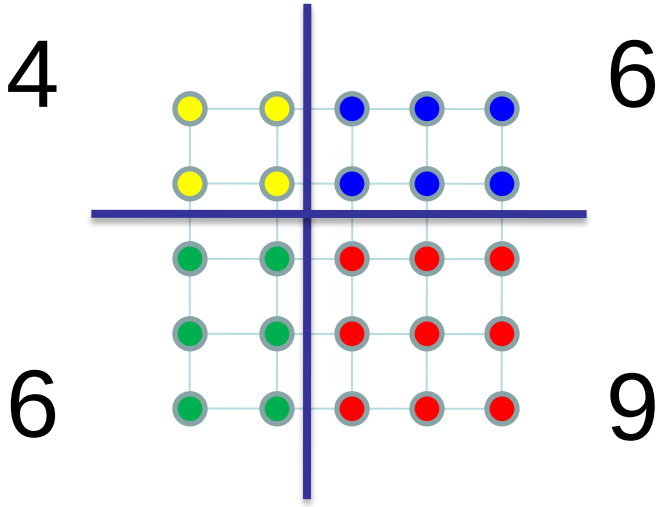
Двойная растровая кривая

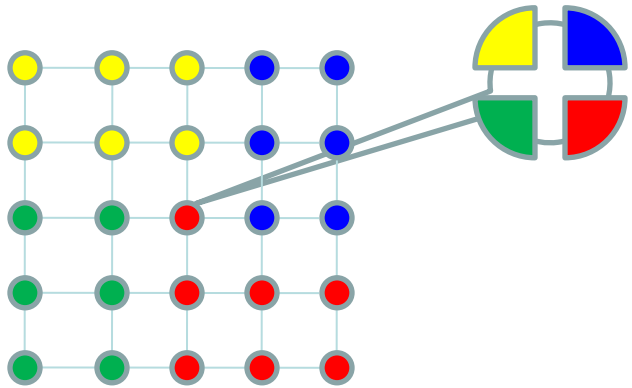


Растровые блоки

Рис. [15]. Кривые, заполняющие пространство

Декомпозиция решетки 5 x 5 на 4 домена

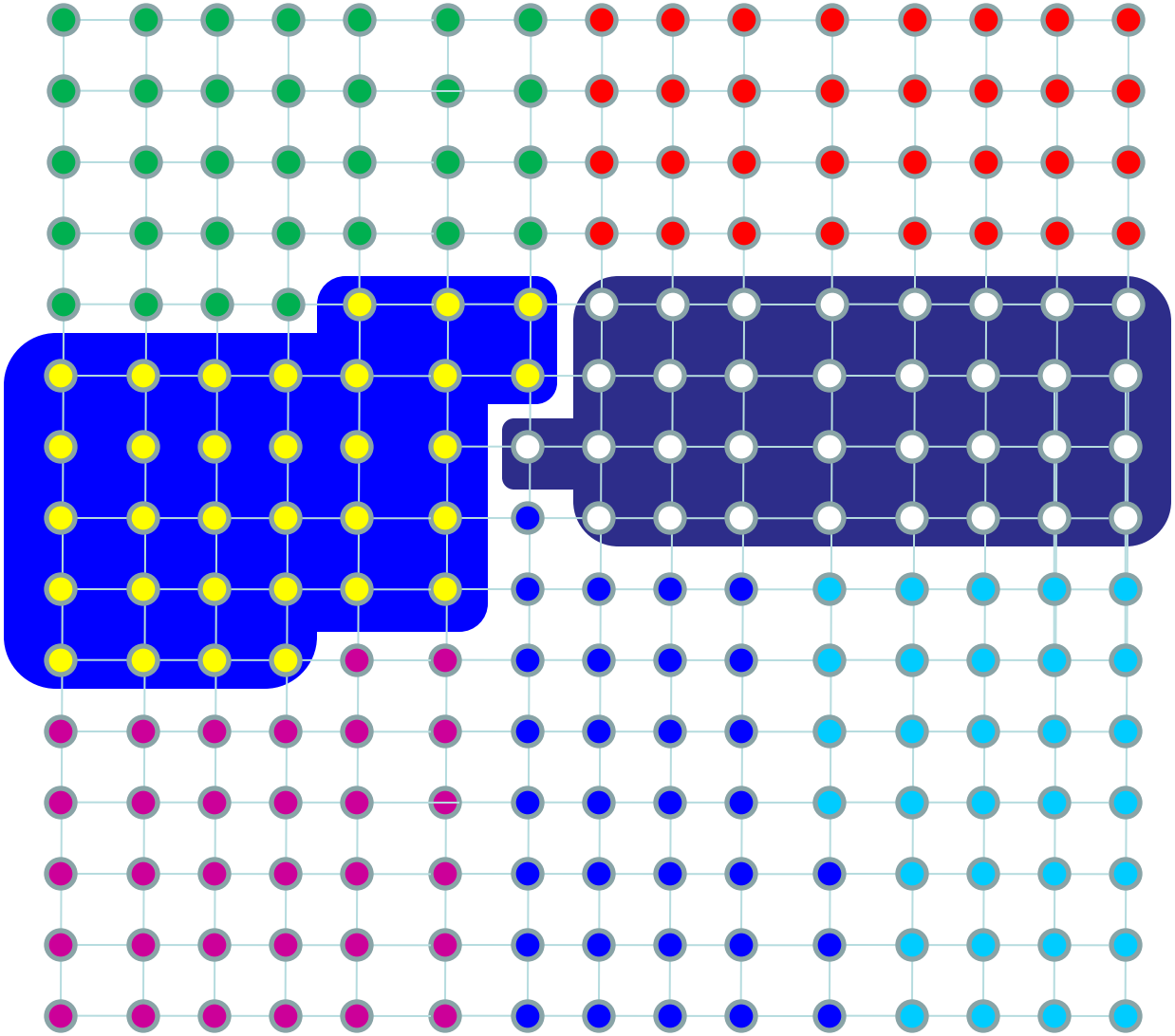


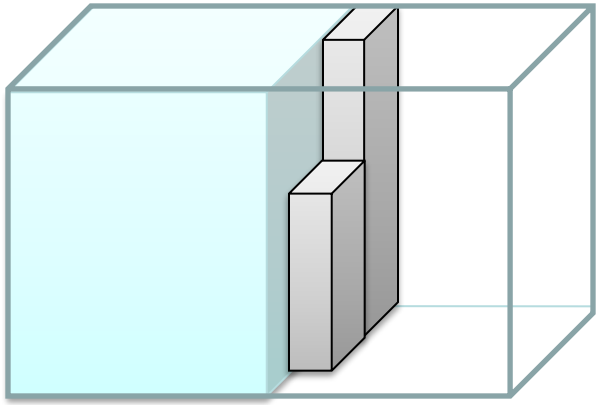


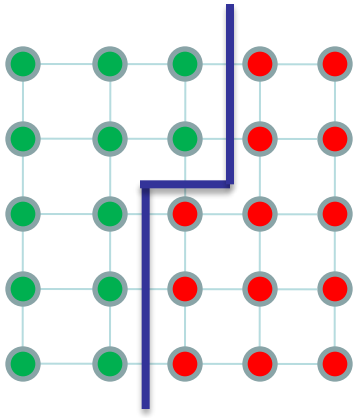
Потери 9/6.25=1.44

Потери 7/6.25=1.12

Декомпозиция сетки 25x25 на 7 частей

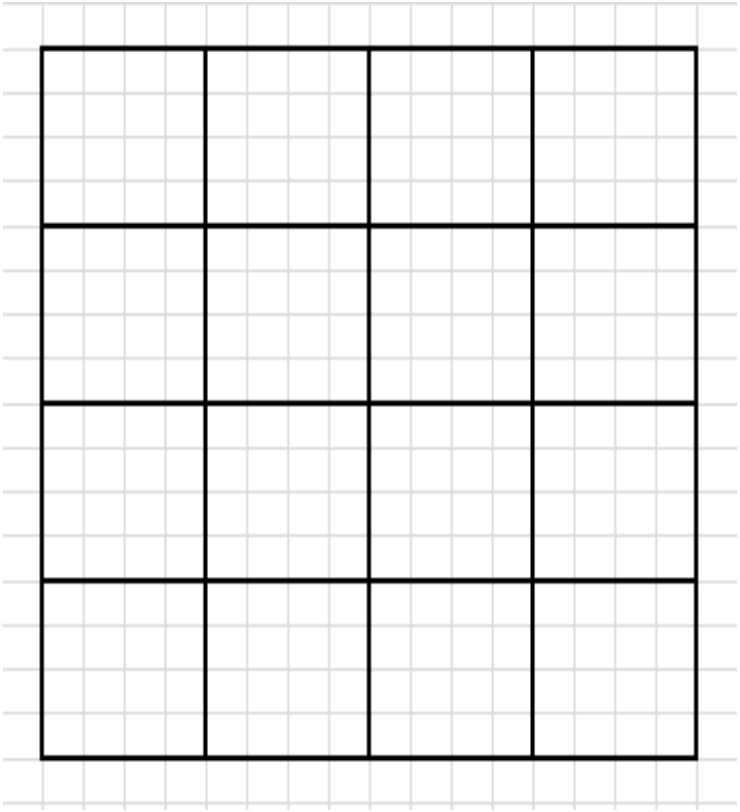






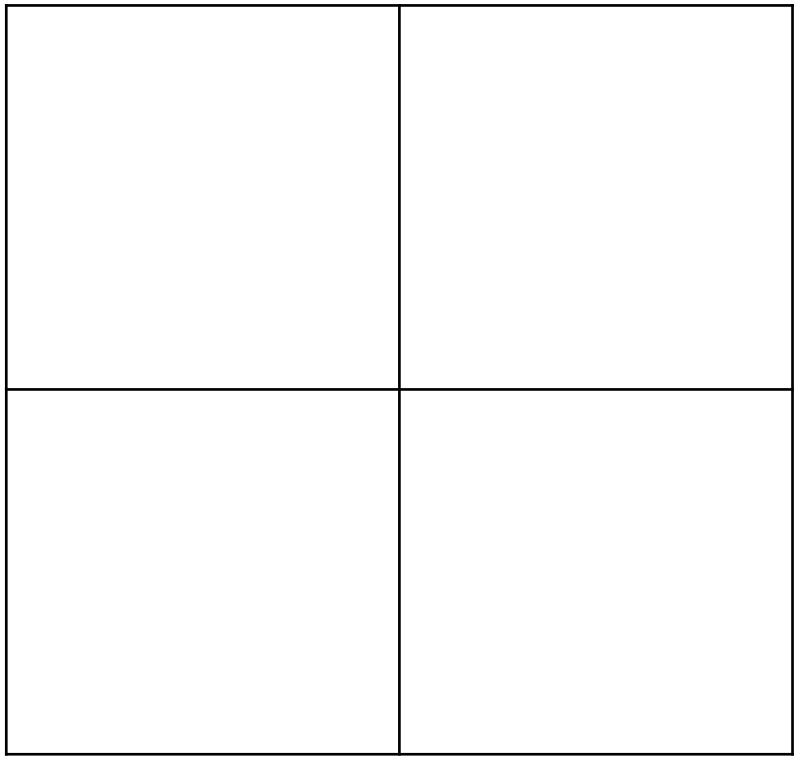
Локально измельчающаяся сетка

- Пусть в начале сетка задана регулярной решёткой базовых ячеек

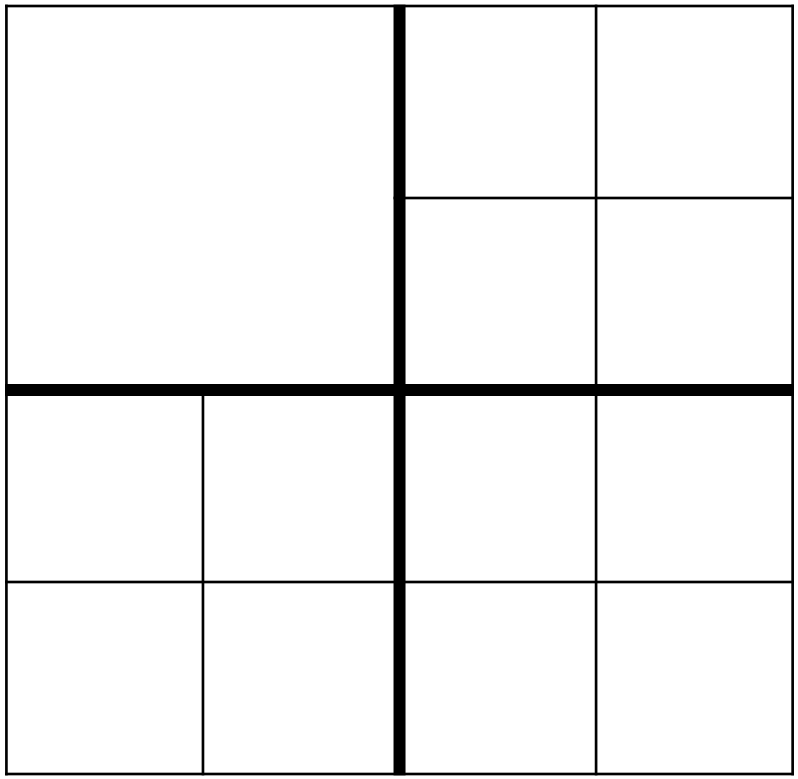


Локально измельчающаяся сетка

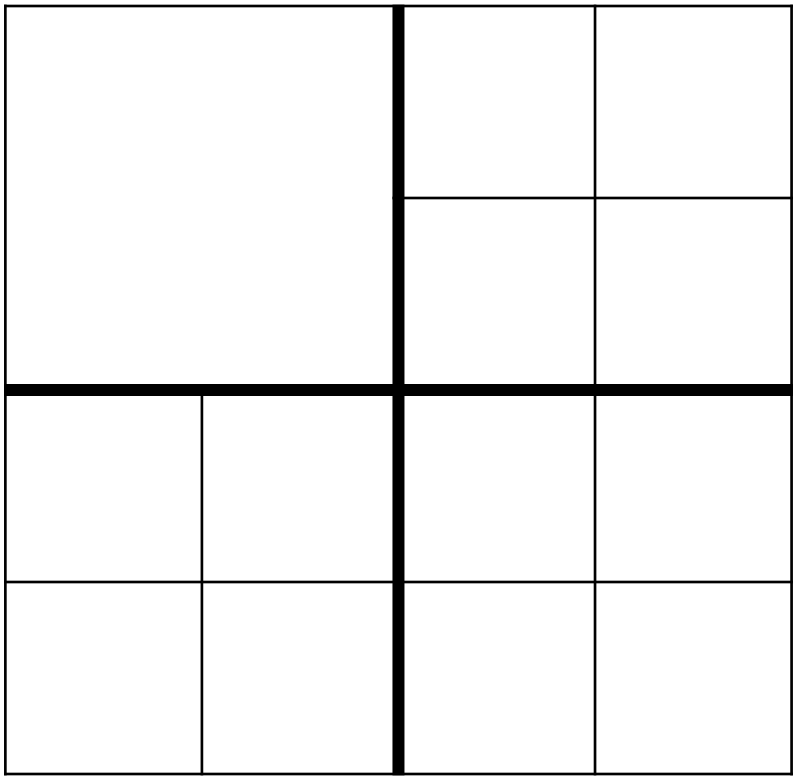
- Исходная сетка задана регулярной решёткой базовых ячеек



Два варианта измельчения базовых ячеек

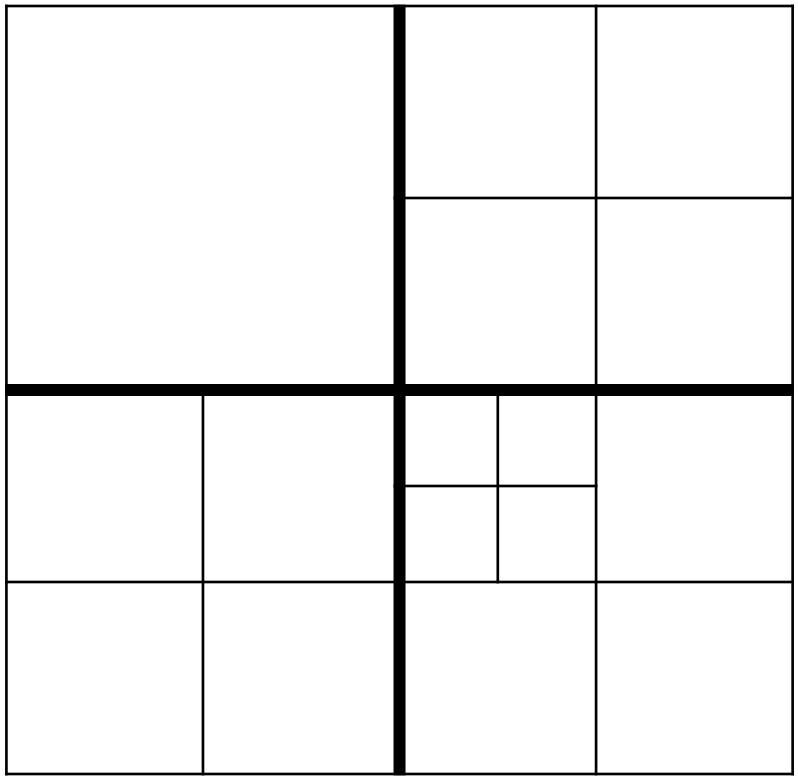


Иерархическое измельчение

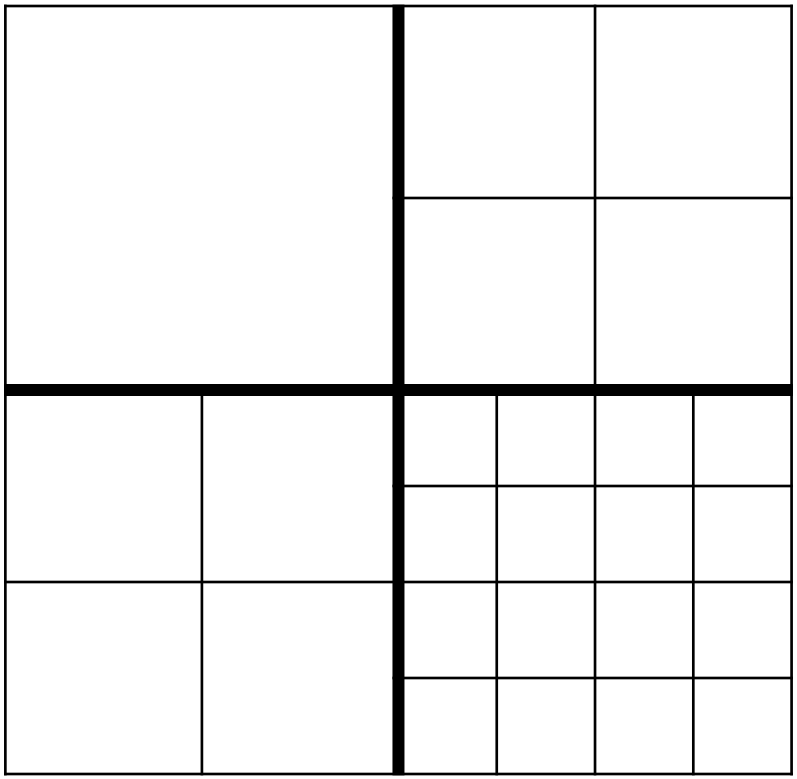


Равномерное измельчение

Два варианта измельчения базовой ячейки

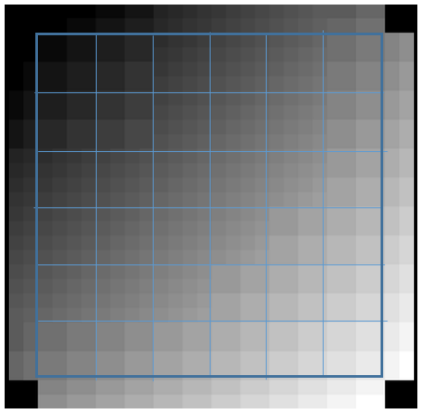
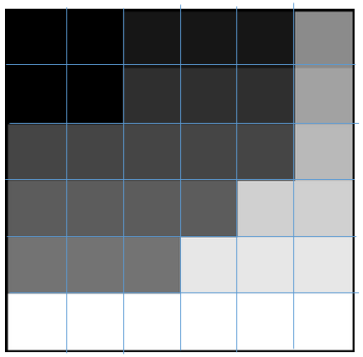
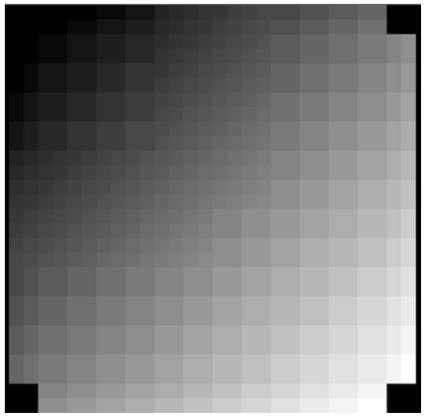
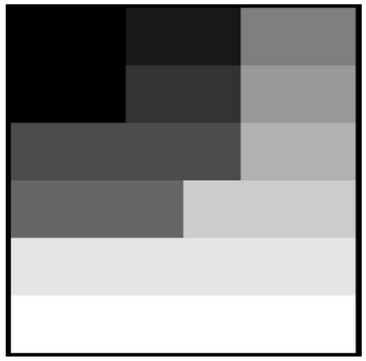
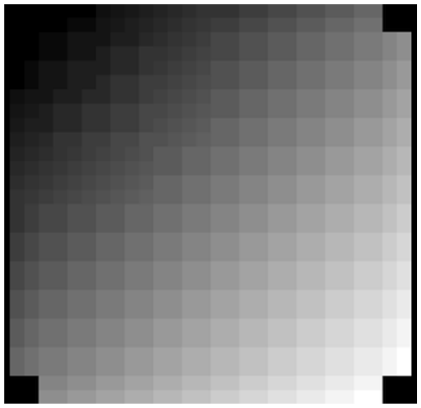
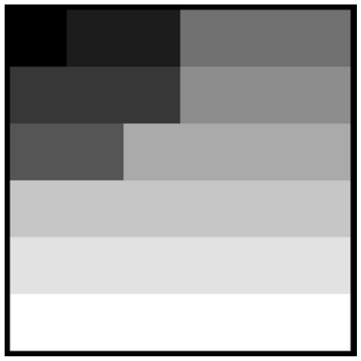
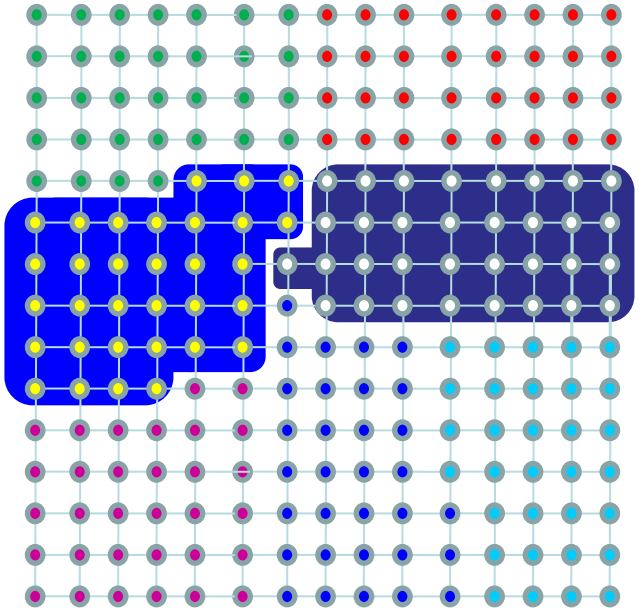


Иерархическое измельчение



Равномерное измельчение

Формирование блоков

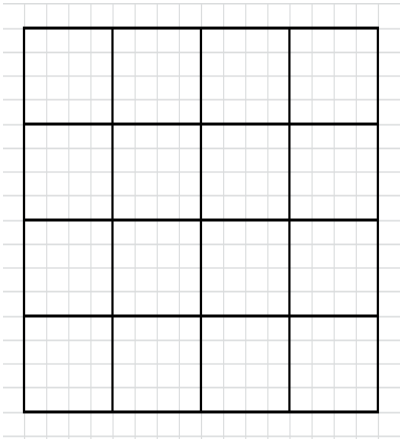
- Блоки одномерны в 2D и в 3D случае
- Время формирования одномерных блоков пропорционально числу базовых ячеек

Потери за счёт излишнего измельчения (%)

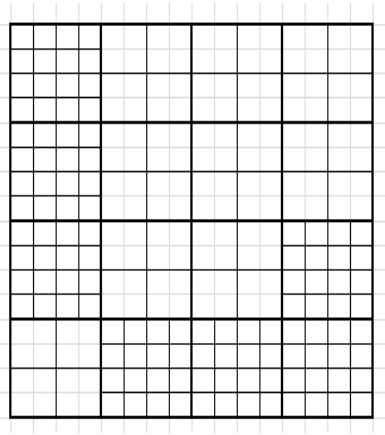
	2D					3D				
1	2	3	4	5	n	1	2	3	4	5
0.57	0.8	1.1	0.85	1.1	100	1.4	2.4	2.8	2.9	3
0.28	0.49	0.55	0.52	0.52	200	0.7	1.2	1.4	1.4	1.4
0.17	0.32	0.39	0.35	0.44	300	0.47	0.83	0.95	0.98	0.99
0.13	0.24	0.28	0.3	0.33	400	0.35	0.62	0.71	0.73	0.74

Равномерное измельчение базовых ячеек

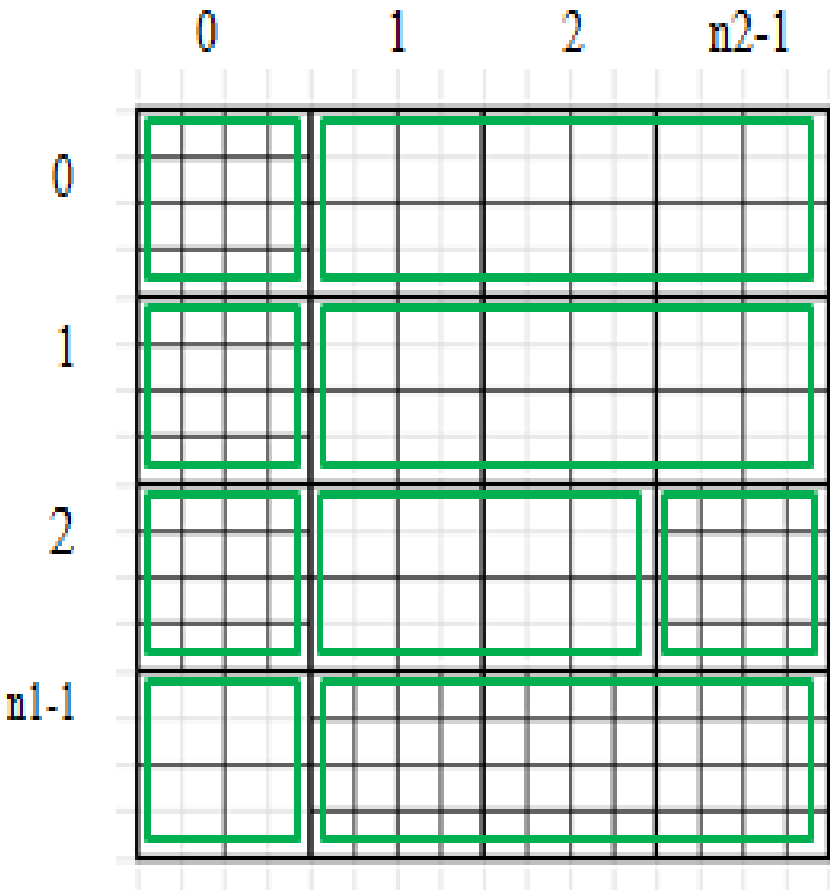
Исходная сетка базовых ячеек



Измельченные в 2 и в 4 раза базовые ячейки

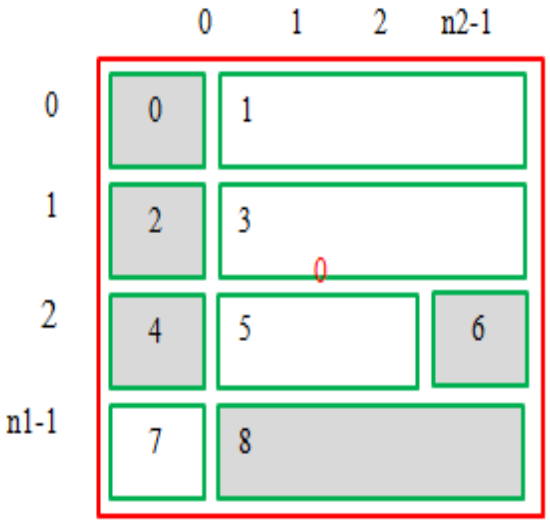


Логические блоки базовых ячеек

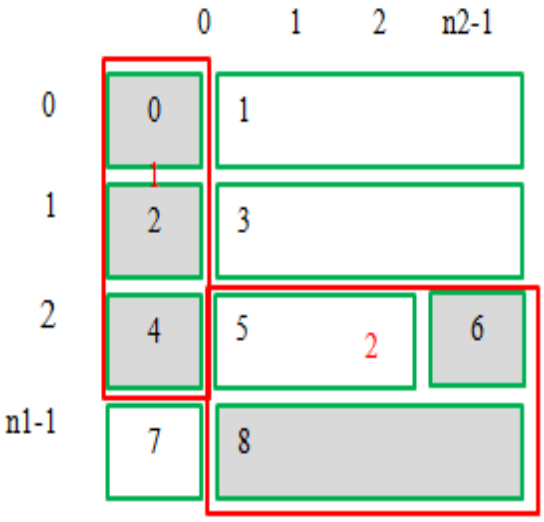


Блоки хранения листовых ячеек

1. Блоки базовых ячеек ориентированы горизонтально
2. Блоки одного уровня измельчения могут быть расположены в одном и том же, или в разных блоках хранения листовых ячеек
3. На основе блоков базовых ячеек формируются задания для выполнения вычислений и передачи данных

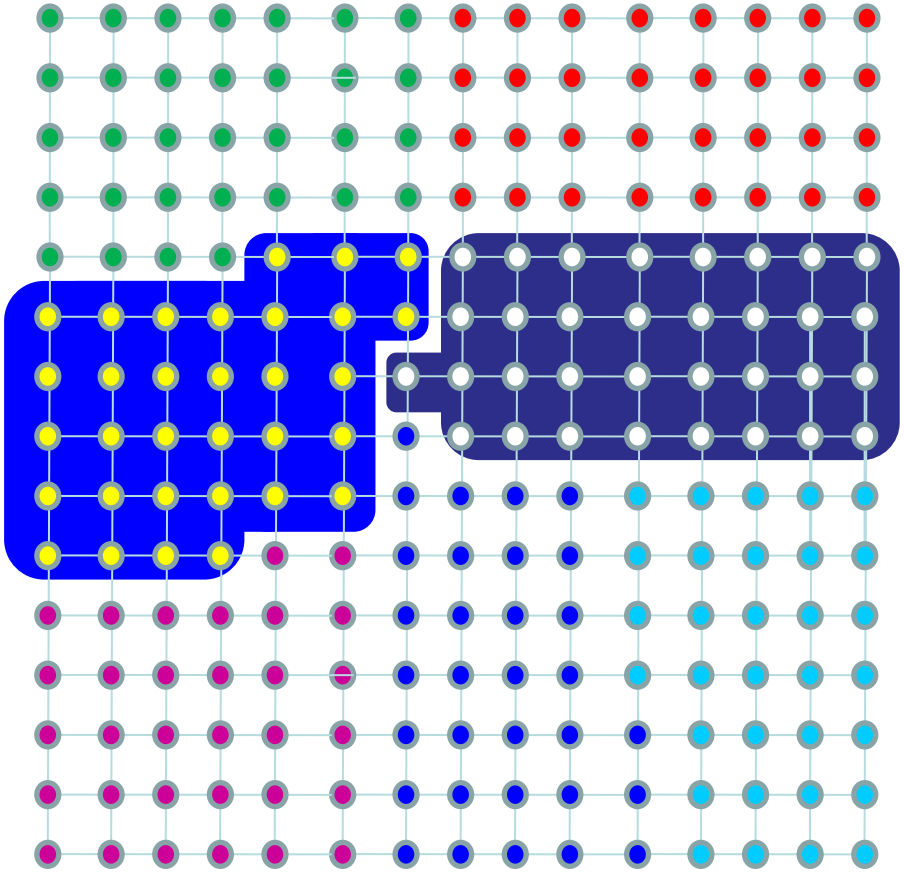


Блок хранения ячеек уровня измельчения 1

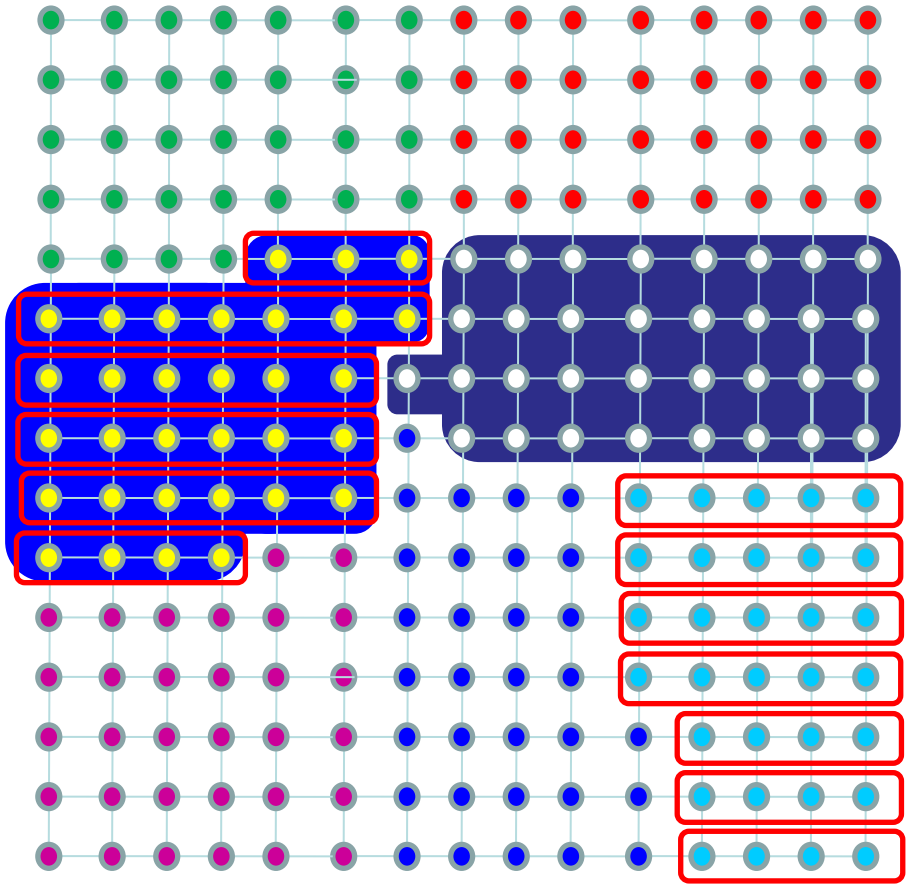


Блоки хранения ячеек уровня измельчения 2

Домен – фрагмент решётки с границей сложной формы



Домен – набор блоков базовых ячеек



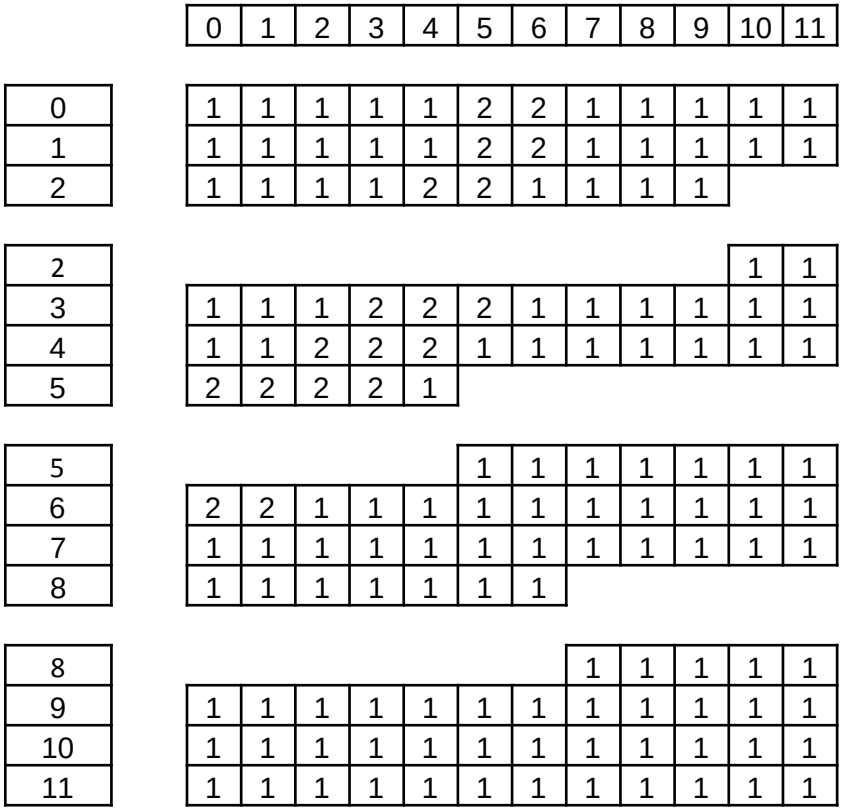
Элементарные задания выполняются над блоками базовых ячеек

- Описывают регулярные вычисления (копирование, интерполяцию или передачу данных) над прямоугольными блоками элементарных ячеек или над их прямоугольными фрагментами
- Каждое элементарное задание может быть назначено на одно или на несколько процессорных ядер или на графический ускоритель
- Содержательная часть программы всегда оперирует прямоугольными блоками. Остальные операции выполняются на уровне библиотеки поддержки
- Упрощается статическая и динамическая балансировка загрузки

Декомпозиция 2D(3D) расчётной сетки выполняется в два(три) этапа – обеспечивается устойчивая декомпозиция

Сетка процессоров p = p1 * p2 : 16 = 4 * 4

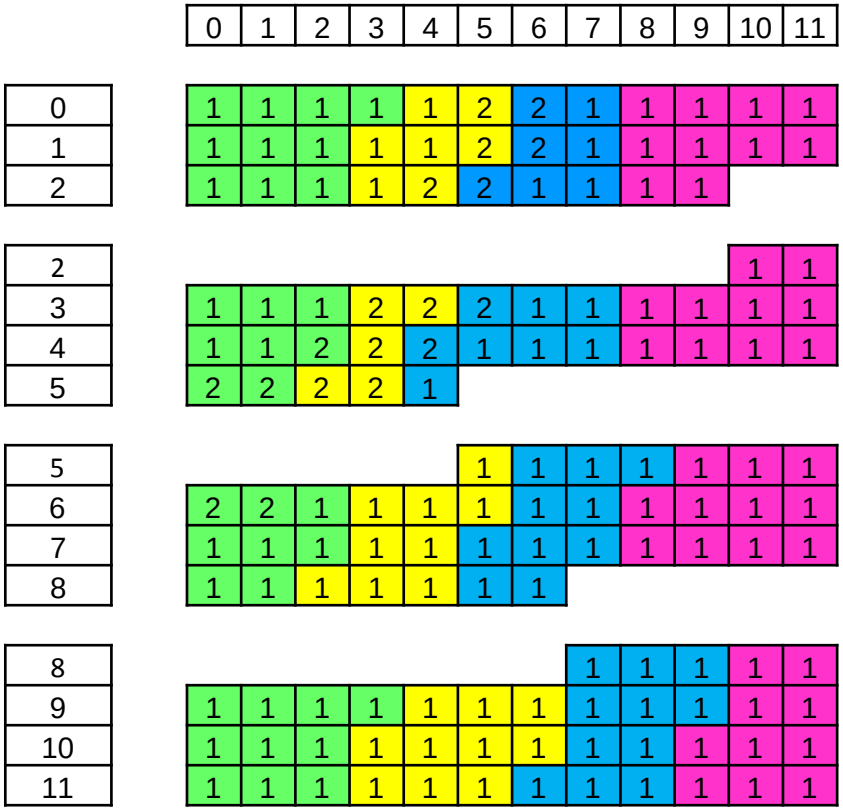
1) Разбиение расчётной области на горизонтальные слои равной трудоёмкости



Декомпозиция 2D(3D) расчётной сетки выполняется в два(три) этапа, что обеспечивает устойчивость декомпозиции

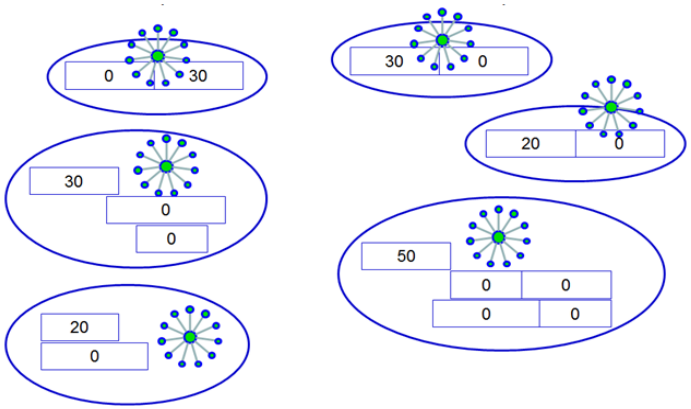
Сетка процессоров p = p1 * p2 : 16 = 4 * 4

1) Разбиение расчётной области на горизонтальные слои равной трудоёмкости
2) Разбиение слоёв на домены



Послойная декомпозиция - устойчивая

- Низкая вычислительная сложность
- Возможность вычисления на каждом процессоре нужного домена и его соседей без вычисления всех доменов сетки
- Разная балансировка на разных этапах



10	10	10	10
11	10	10	10
10	9	10	11
10	10	11	10

Рис. [18]. Однородный расчёт

0	1	2	3	4	5	6	7	8	9	10	11
0	1	1	1	1	1	2	2	1	1	1	1	1
1	1	1	1	1	1	2	2	1	1	1	1	1
2	1	1	1	1	2	2	1	1	1	1		
2											1	1
3	1	1	1	2	2	2	1	1	1	1	1	1
4	1	1	2	2	2	1	1	1	1	1	1	1
5	2	2	2	2	1							
5						1	1	1	1	1	1	1
6	2	2	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1					
8								1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1

0	30	30	0
30	50	20	0
20	0	0	0
0	0	0	0

Рис. [19]. Динамическая балансировка

Автоматизация программирования для перспективных высокопроизводительных вычислительных комплексов

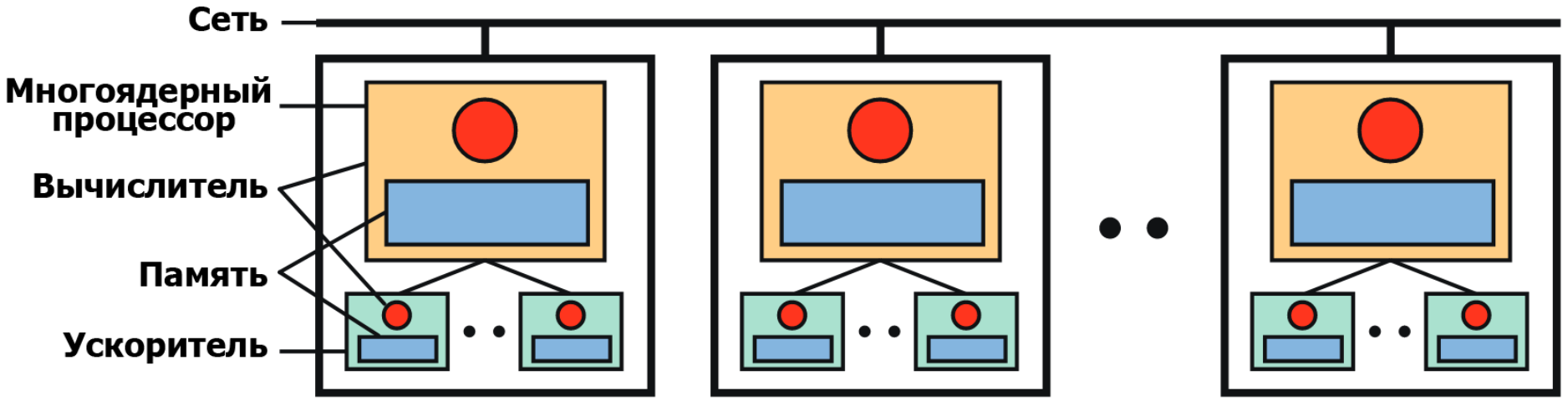
В.А. Крюков, В.А. Бахтин, …

DVMH-программы:

Эффективное выполнение одной и той же программы:

- кластеры различной архитектуры
- многоядерные универсальные процессоры
- графические ускорители
- сопроцессоры Intel Xeon Phi
- …

DVMH-модель параллелизма



DVMH-модель позволяет создавать эффективные параллельные программы для гетерогенных вычислительных кластеров, в узлах которых в качестве вычислительных устройств (вычислителей) наряду с универсальными многоядерными процессорами могут использоваться ускорители различной архитектуры (графические процессоры, сопроцессоры Intel Xeon Phi, ...)

При этом отображенные на узел вычисления автоматически распределяются между вычислителями узла с учетом их производительности

Система автоматизации разработки параллельных программ (DVM-система)

- Компиляторы с языков Fortran-DVMH и C-DVMH

  преобразование последовательной программы в параллельную, использующую стандартные технологии программирования MPI, OpenMP и CUDA

- Библиотека поддержки LIB-DVMH

  реализация модели выполнения параллельной программы; динамическая настройка DVMH-программ на выделенные для их выполнения ресурсы: множество узлов кластера, ядер, ускорителей с учетом их производительности

- DVMH-отладчик

  автоматизированная отладка параллельных программ с использованием методов сравнительной отладки и динамического контроля корректности

- Анализатор производительности DVMH-программ

  получение информации об основных характеристиках эффективности выполнения параллельной программы и ее частей на вычислительной системе

Средства программирования

C-DVMH = Язык Си(C++) + специальные прагмы

Fortran-DVMH = Язык Фортран 95 + специальные комментарии

- Специальные комментарии и прагмы являются высокоуровневыми спецификациями параллелизма в терминах последовательной программы
- Отсутствуют низкоуровневые передачи данных и синхронизации
- Последовательный стиль программирования
- Спецификации параллелизма «невидимы» для стандартных компиляторов
- Существует единственный исходный код программы: один и для последовательного и для параллельного выполнения

Алгоритм Якоби в модели DVMH

```
FILE *f;
#pragma dvm array distribute[block][block] shadow[1:1][1:1]
double A[L][L];
#pragma dvm array align ([i][j] with A[i][j])
double B[L][L];
int main(int argc, char *argv[]) {
    for(int it = 0; it < ITMAX; it++) {
        #pragma dvm region inout (A,B)
        {
            #pragma dvm parallel([i][j] on A[i][j])
            for (int i = 1; i < L - 1; i++)
                for (int j = 1; j < L-1; j++)  A[i][j] = B[i][j];
            #pragma dvm parallel([i][j] on B[i][j]), shadow_renew(A)
            for (int i = 1; i < L - 1; i++)
                for (int j = 1; j < L - 1; j++)
                    B[i][j] = (A[i - 1][j] + A[i + 1][j] + A[i][j - 1] + A[i][j + 1]) / 4.;
        }
    }
   f = fopen("jacobi.dat", "wbsl");
   #pragma dvm get_actual(B)
   fwrite(B, sizeof(double), L * L, f);
   fclose(f);
   return 0;
}
```

Система Автоматизации расПараллеливания ФОРтран программ (САПФОР)



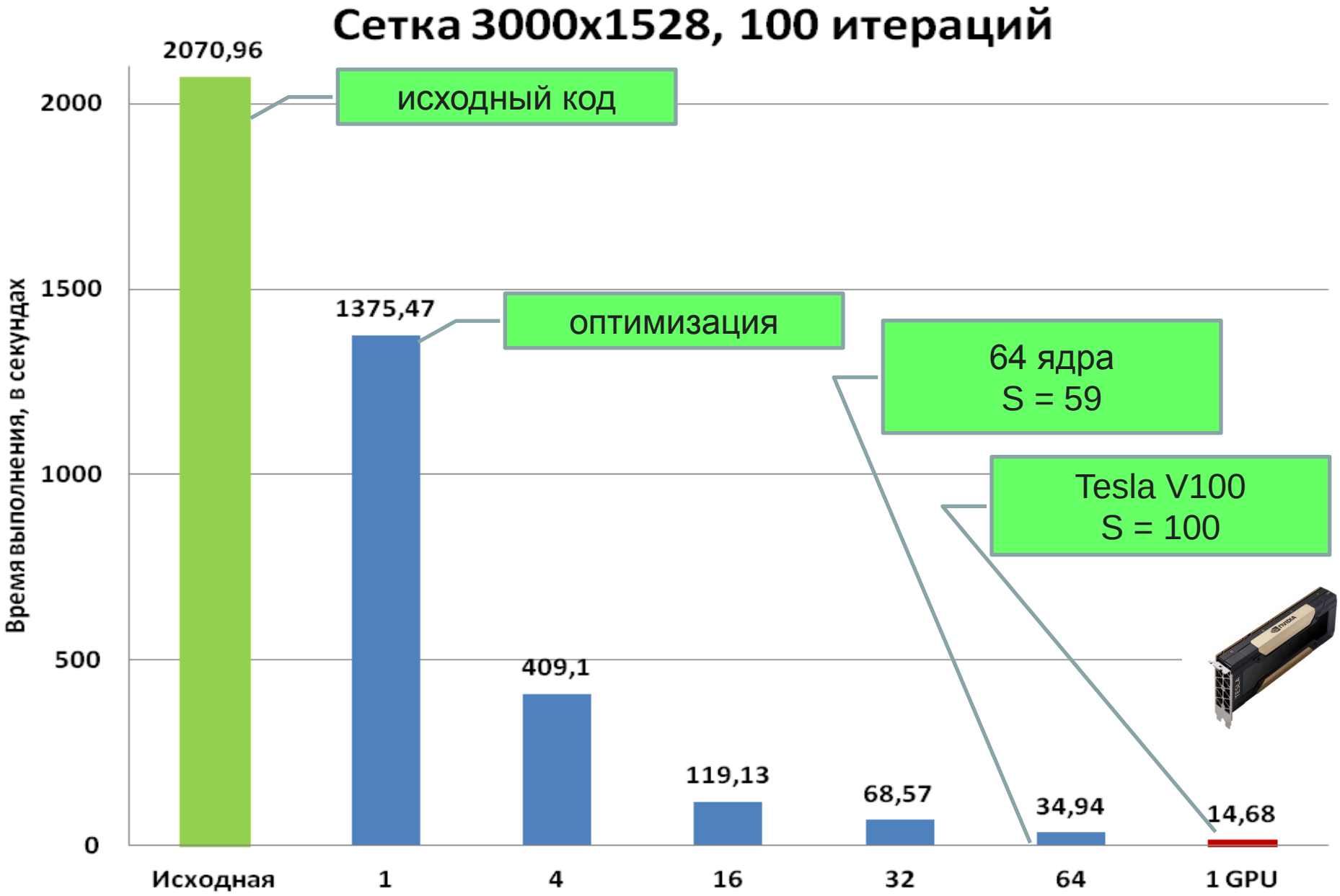
Последовательная программа

Последовательная программа
+ свойства программы
+ преобразования программы

САПФОР

Параллельная программа в модели DVMH

Время выполнения программы на разном числе ядер и на графическом ускорителе



1 vs 4
NVidia Tesla V100

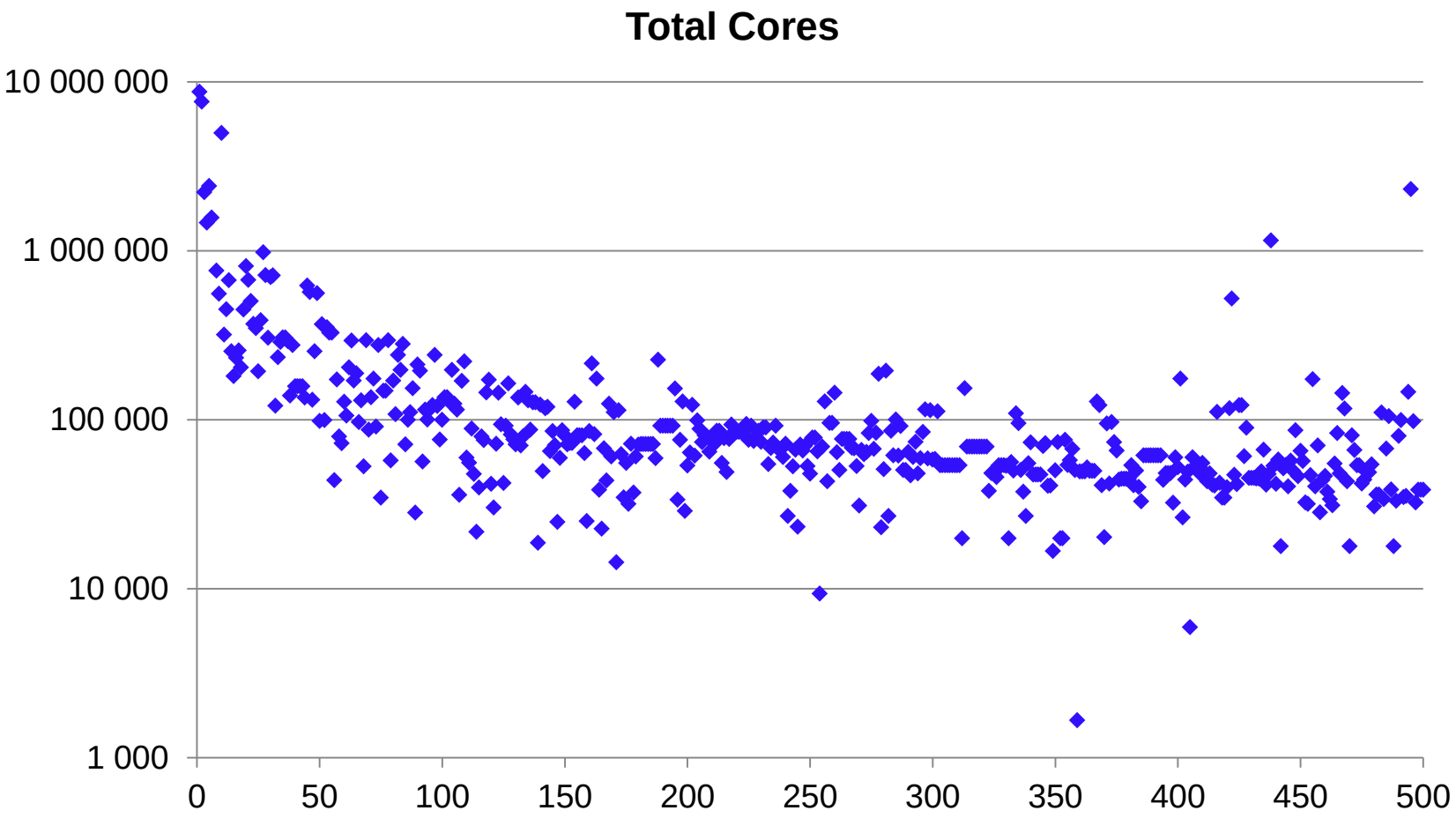
Сетка 120 000 x 5 028

- Накладные расходы 0.95 -> 5.64 с
- Общее время 162 -> 48 с

Ускорение = 3.3



Число вычислительных ядер суперкомпьютеров top-500 2022.11



ВОЗМОЖНОСТИ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ

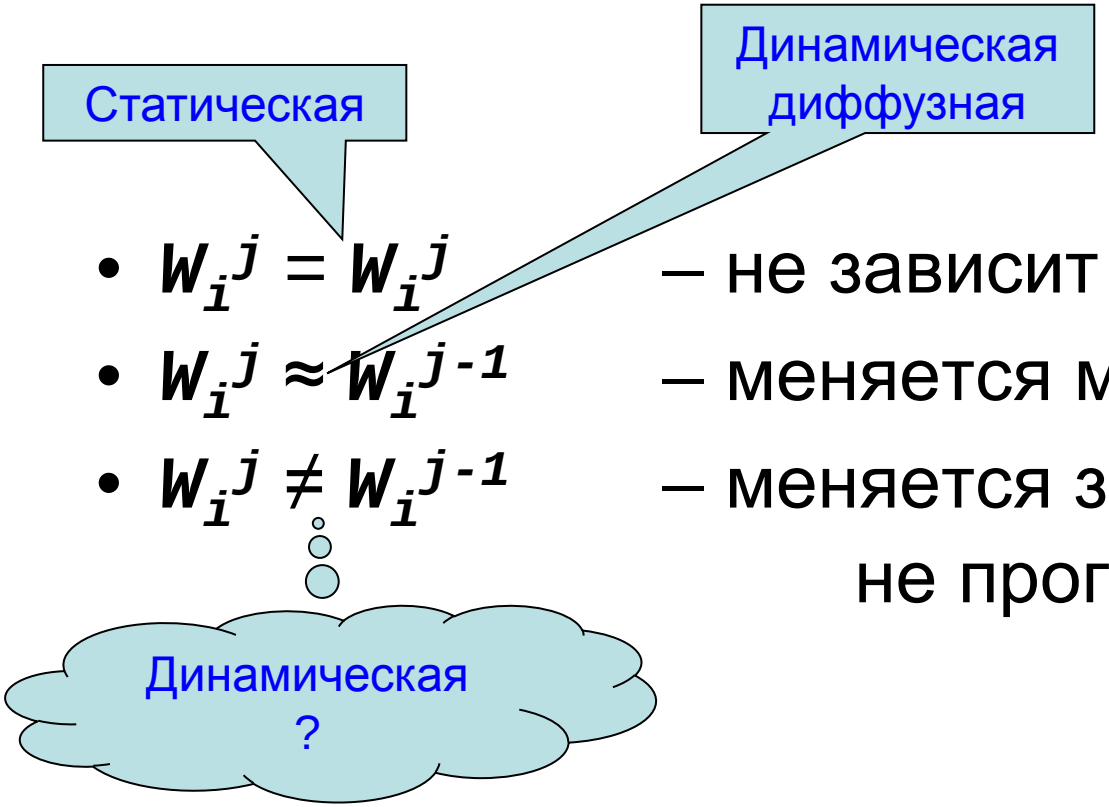
- **Высокопроизводительные вычисления позволяют значительно ускорить (в 2 - 2.5 раза) и удешевить создание новой техники и новых технологий**

- **Они являются залогом национальной безопасности России, обеспечивая конкурентоспособность практически во всех направлениях фундаментальной науки и в ключевых областях народно-хозяйственной деятельности**

- **Для эффективного применения суперкомпьютеров необходима разработка качественно новых, эффективных на всех уровнях выполнения, параллельных алгоритмов и поддерживающих их разработку инструментальных средств**

Стратегии балансировки загрузки

W_i^j - вычислительная нагрузка, ассоциированная с узлом сетки i на шаге j



- $W_i^j = W_i^j$ – не зависит от времени
- $W_i^j \approx W_i^{j-1}$ – меняется медленно
- $W_i^j \neq W_i^{j-1}$ – меняется значительно и не прогнозируемо

Причины возникновения дисбаланса

- неудачное начальное распределение элементарных заданий
- изменение эффективной производительности процессоров
- изменение трудоемкости обработки элементарных заданий (например, вычислений в узлах сетки) уже в ходе выполнения работы
- Неэффективное управление потоками и контекстами

Свойства элементарных заданий

Статические

• Не всегда можно заранее точно предсказать трудоемкость обработки узлов сетки

Динамические

• Трудоемкость обработки каждого из узлов расчетной сетки может отличаться на разных моментах модельного времени

Свойства процессоров

Статические

- Процессоры могут обладать разной, неизвестной заранее производительностью

Динамические

- Эффективная производительность процессоров может меняться с течением времени

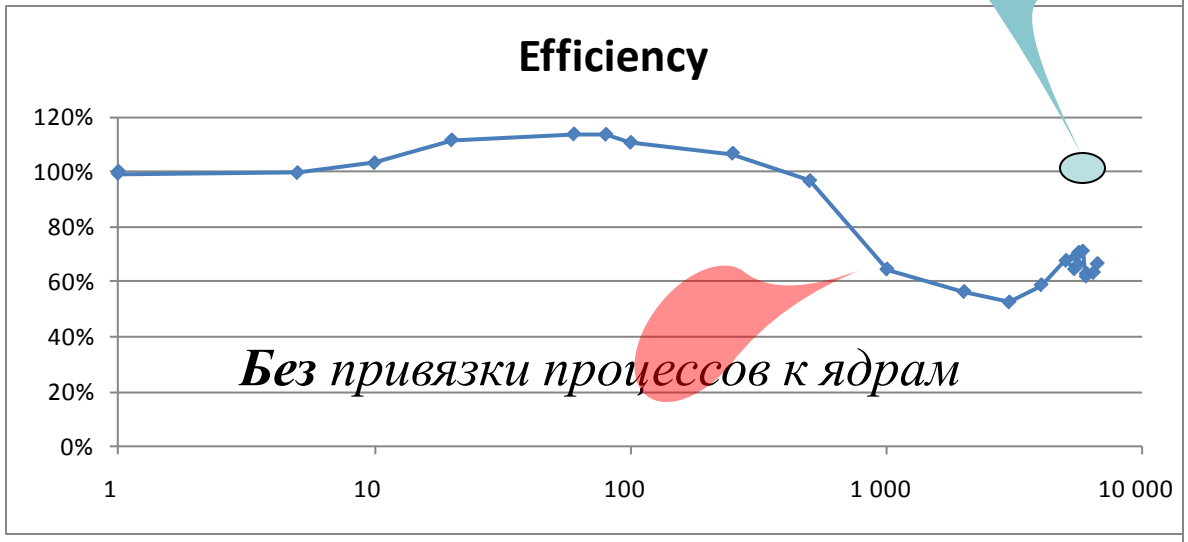
Результаты тестирования производительности

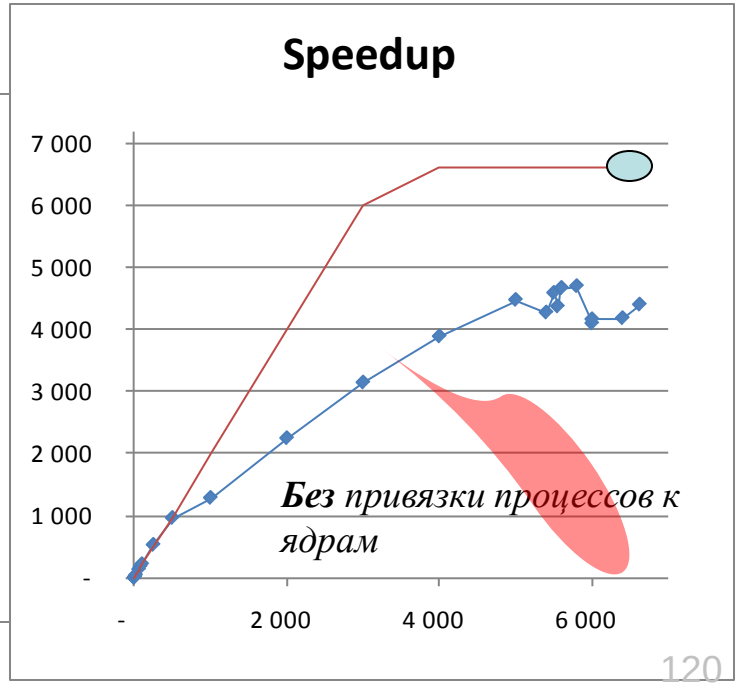
- *асинхронные обмены,*
- *коллективные операции редуцирования*
- ***привязка MPI-процессов к ядрам** процессоров*

*обеспечивают отсутствие деградации эффективности (**99%**) на доступном числе ядер при*

- *компактном шаблоне*
- *сетке 1 миллиард узлов*
- *7728 ядрах*

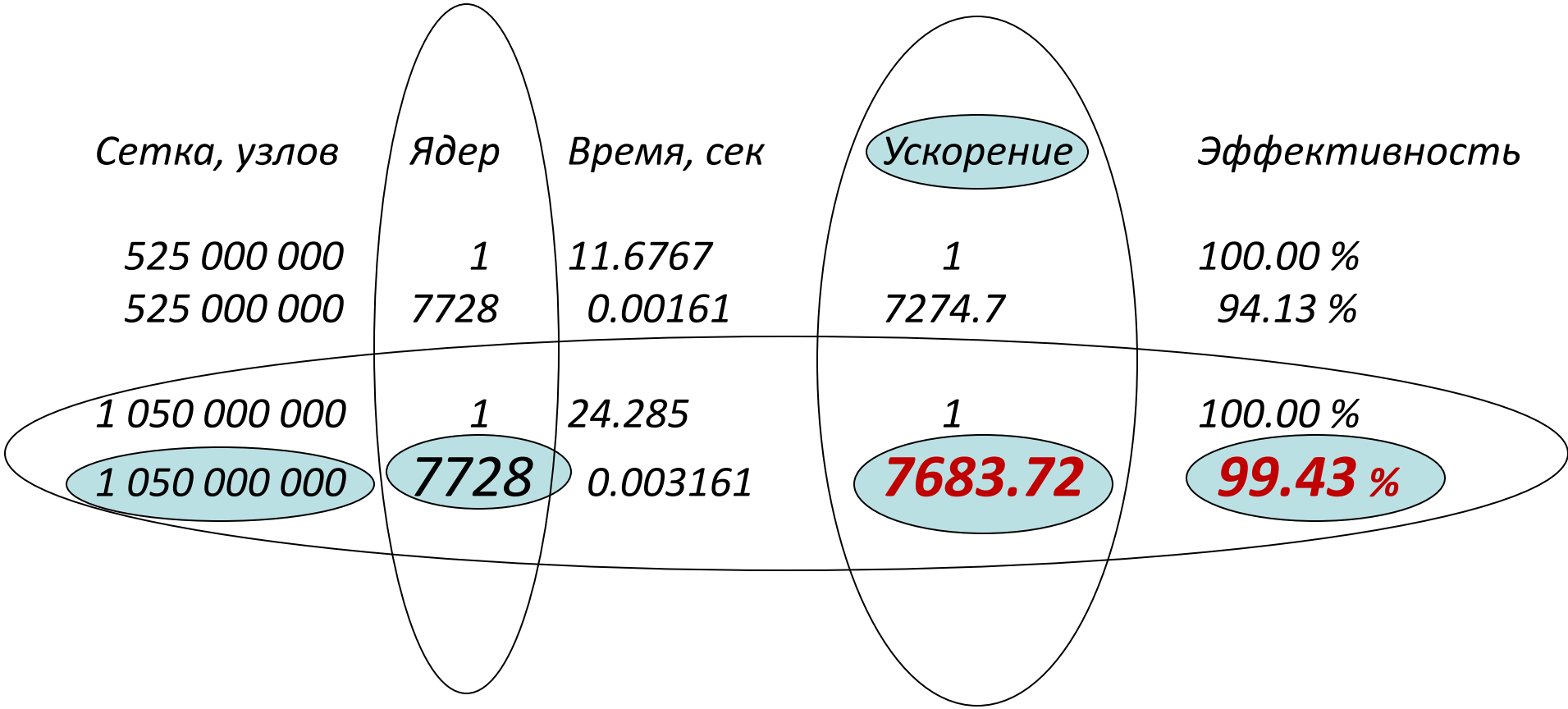
***Без** привязки процессов к ядрам эффективность расчетов на числе процессов, большем чем число узлов, падает до величины порядка 70%.*





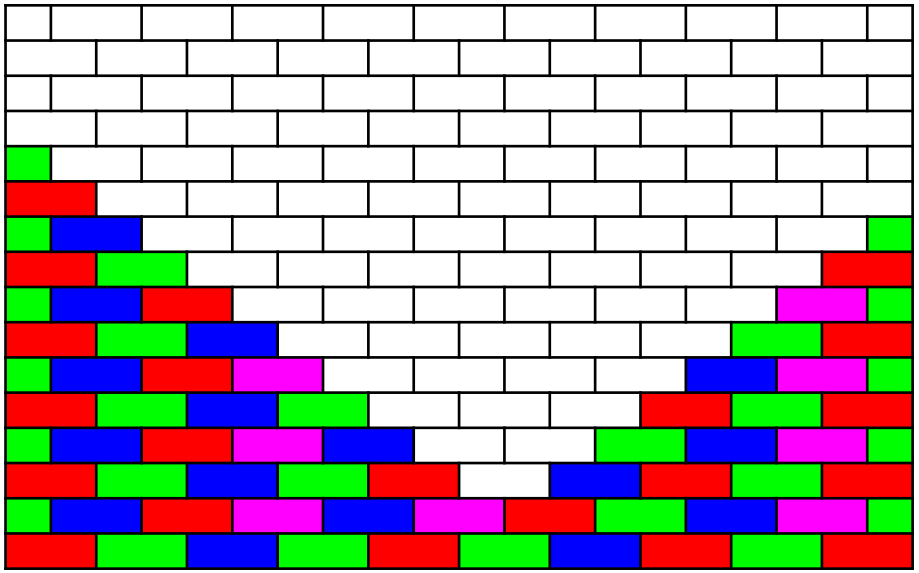
Эффективность расчета 99%

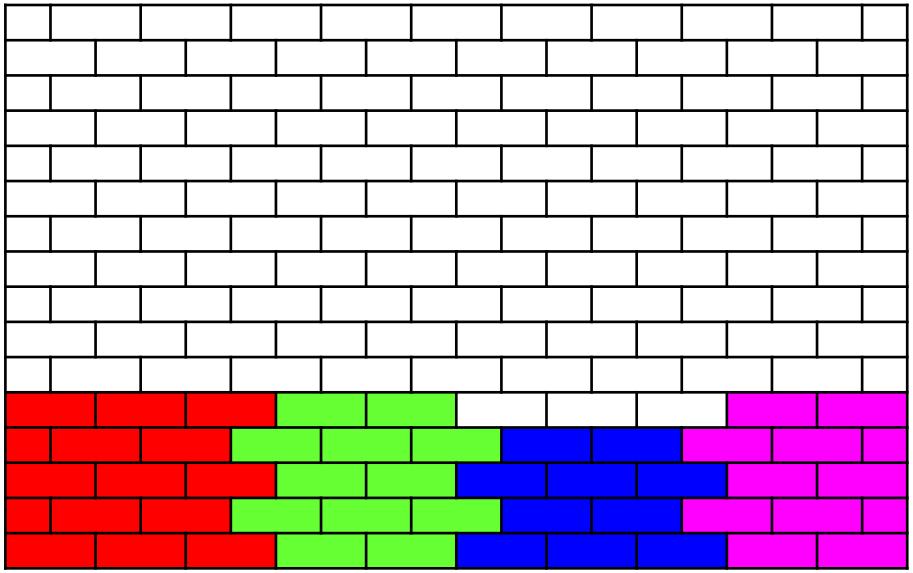
996 узлов 966 по 8 ядер – всего 7728 ядер,
организованных в трёхмерную решетку размером 24 x 161 x 2



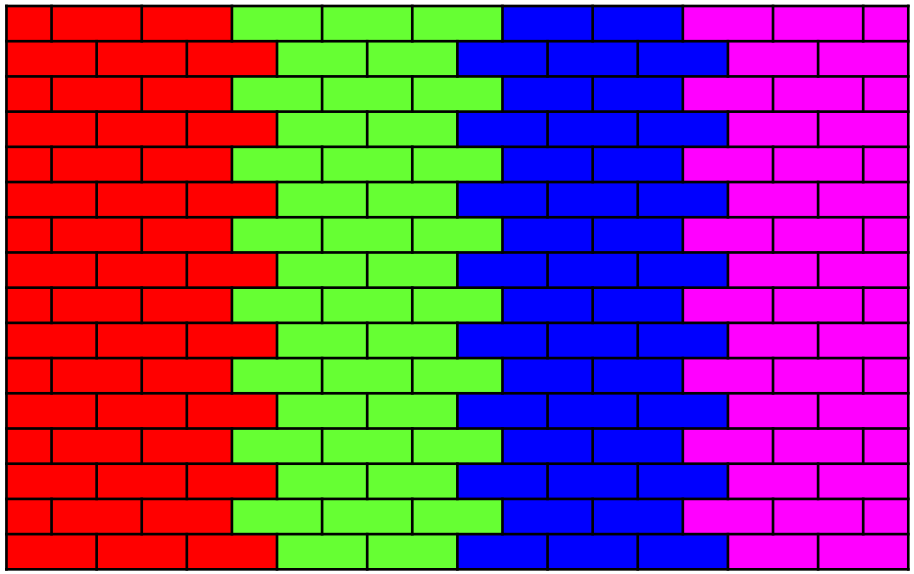
Сетка, узлов	Ядер	Время, сек	Ускорение	Эффективность
525 000 000	1	11.6767	1	100.00 %
525 000 000	7728	0.00161	7274.7	94.13 %
1 050 000 000	1	24.285	1	100.00 %
1 050 000 000	7728	0.003161	**7683.72**	**99.43 %**

Метод геометрического параллелизма





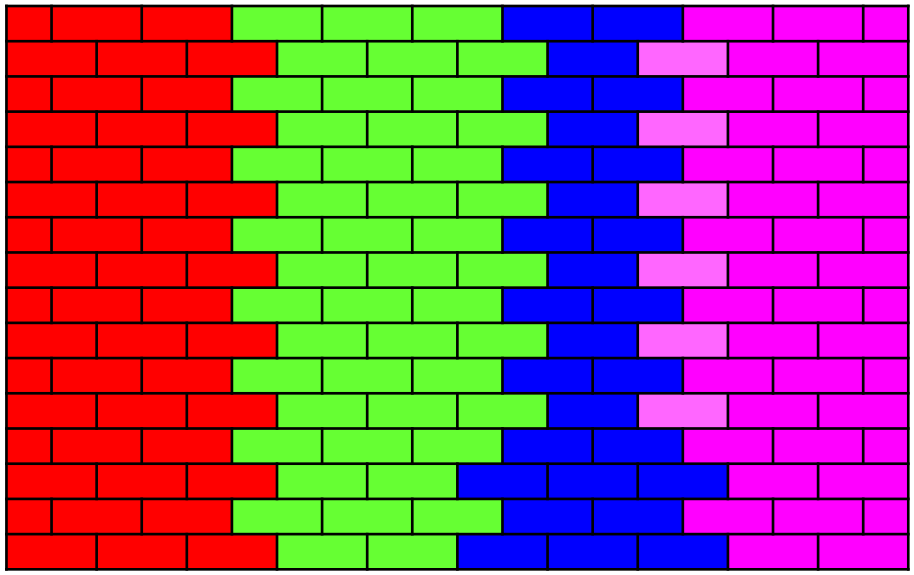
Метод геометрического параллелизма



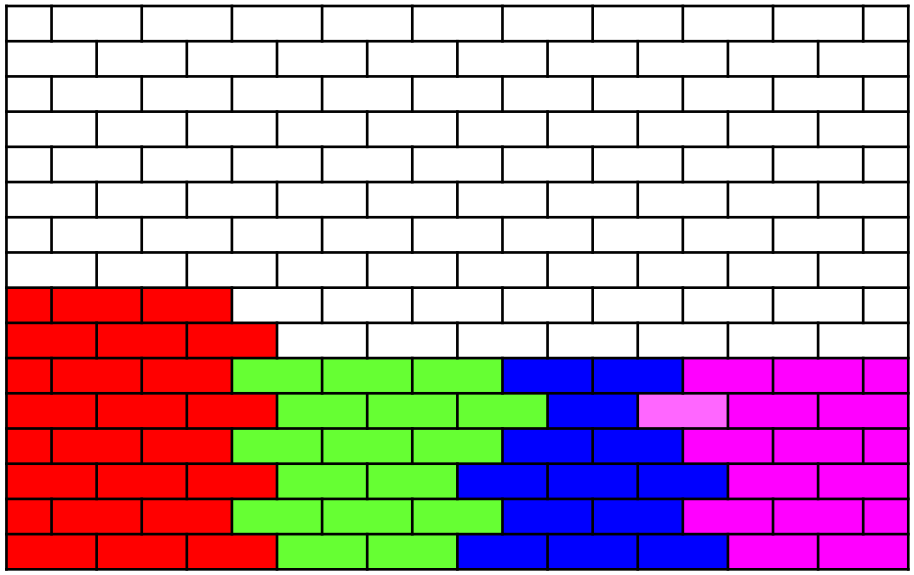
$$T_1(kn) = \tau_c kn \qquad T_p(kn) = \tau_c \frac{kn}{p} + 4k\tau_s$$

$$S_p(kn) = p\frac{1}{1+4\frac{p}{n}\frac{\tau_s}{\tau_c}} \qquad E_p(kn) = \frac{1}{1+4\frac{p}{n}\frac{\tau_s}{\tau_c}}$$

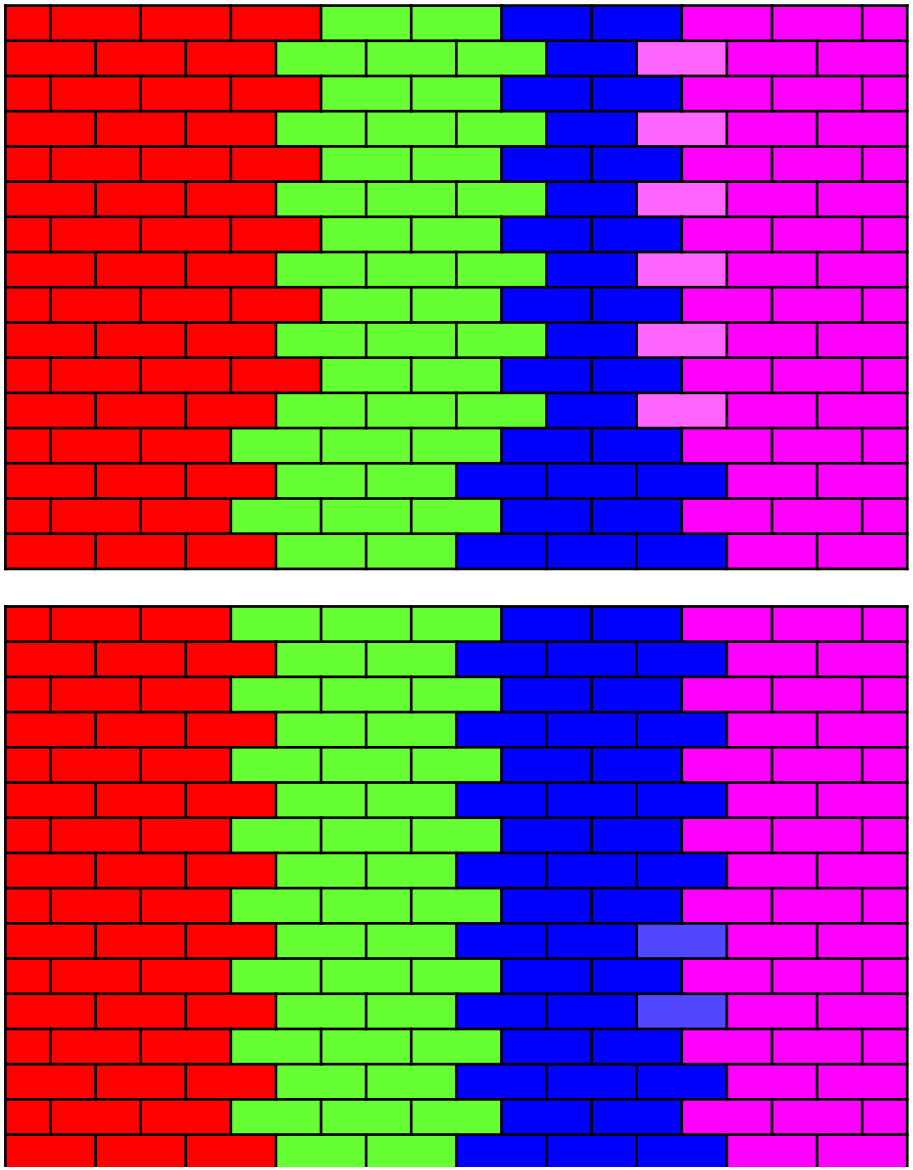
Диффузная балансировка загрузки



Диффузная балансировка загрузки



Диффузная балансировка загрузки



Статическое распределение

если Дисбаланс вычислительной нагрузки обусловлен разными производительностями процессоров

$$n_i' = n \frac{\frac{n_i}{t_i}}{\sum_{j=1}^{p} \frac{n_j}{t_j}}$$

если Дисбаланс обусловлен различной трудоемкостью обработки узлов сетки

процессоры обладают одинаковой производительностью

Общая вычислительная нагрузка определяется как

$$T = \sum_{k=1}^{p} t_i$$

$$T_{mid} = \frac{T}{p}$$

Алгоритм распределения заданий по одинаковым процессорам

```
m=0;    t=0;
for(j=0;j<n;j++)
{
   t=t+σ_j
   if(t ≥ T_mid)    {          n_i''=j-m;    m=j;    t=0;    }
}
```



Каждому процессору назначается множество очередных заданий, суммарная трудоёмкость которых равна средней по всем процессорам

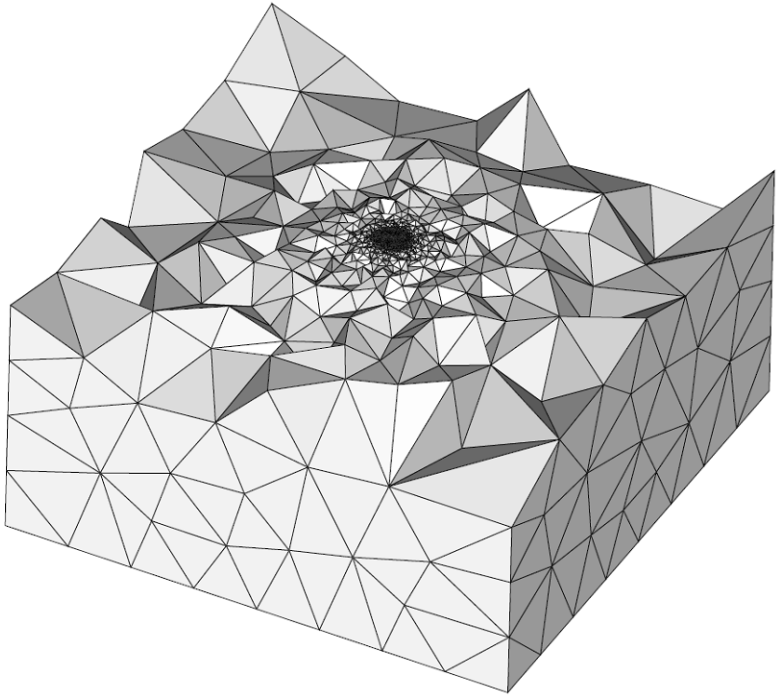


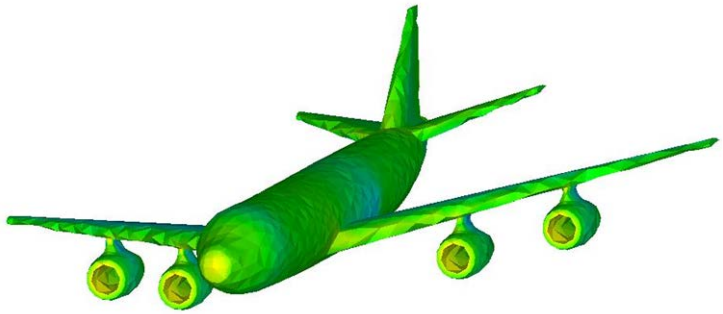
Перераспределение узлов сетки

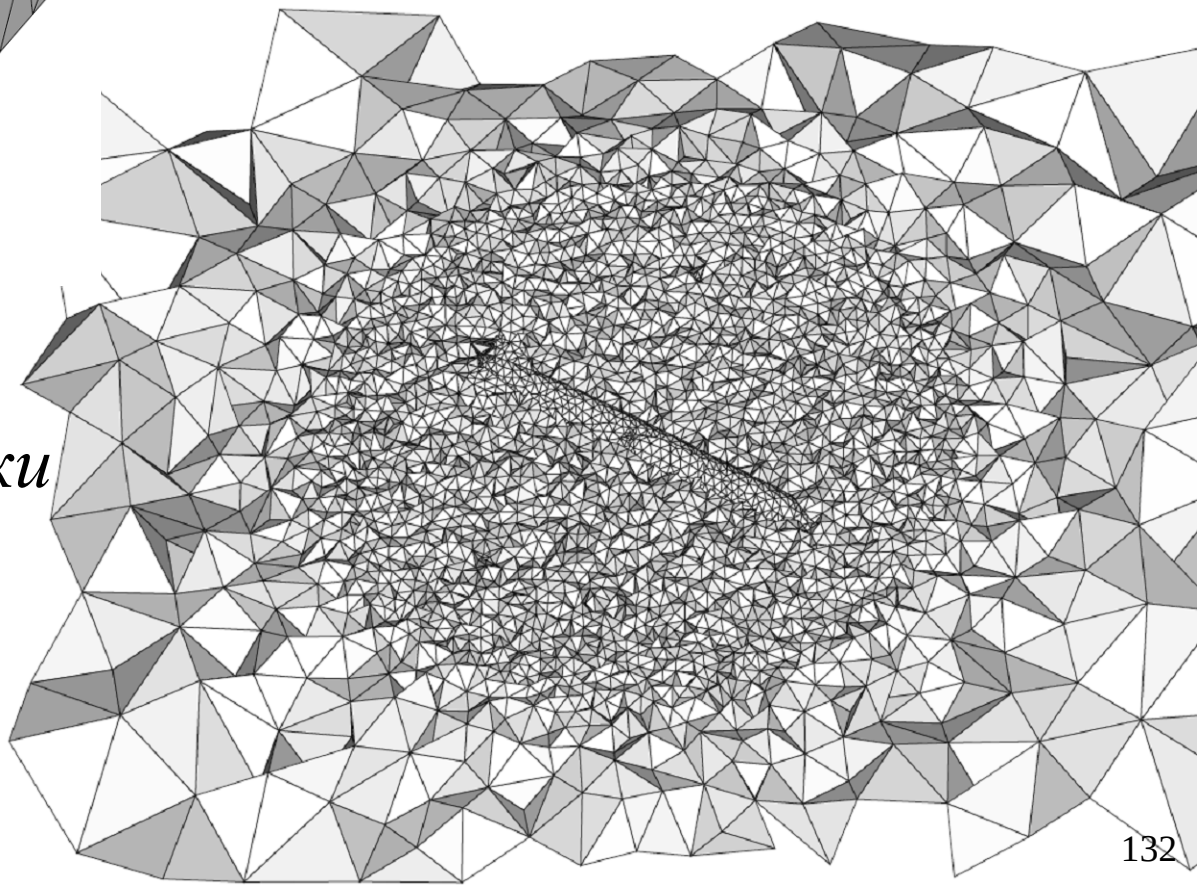
номер i процессора	1	2	3	4
n_i	5	10	14	7
t_i	1	2	1	3
n_i'	7	7	19	3
n_i''	10	10	11	5

Два разных решения
Какое верно**?**
Почему одно из них верно**?**

Сетки Неструктурированные

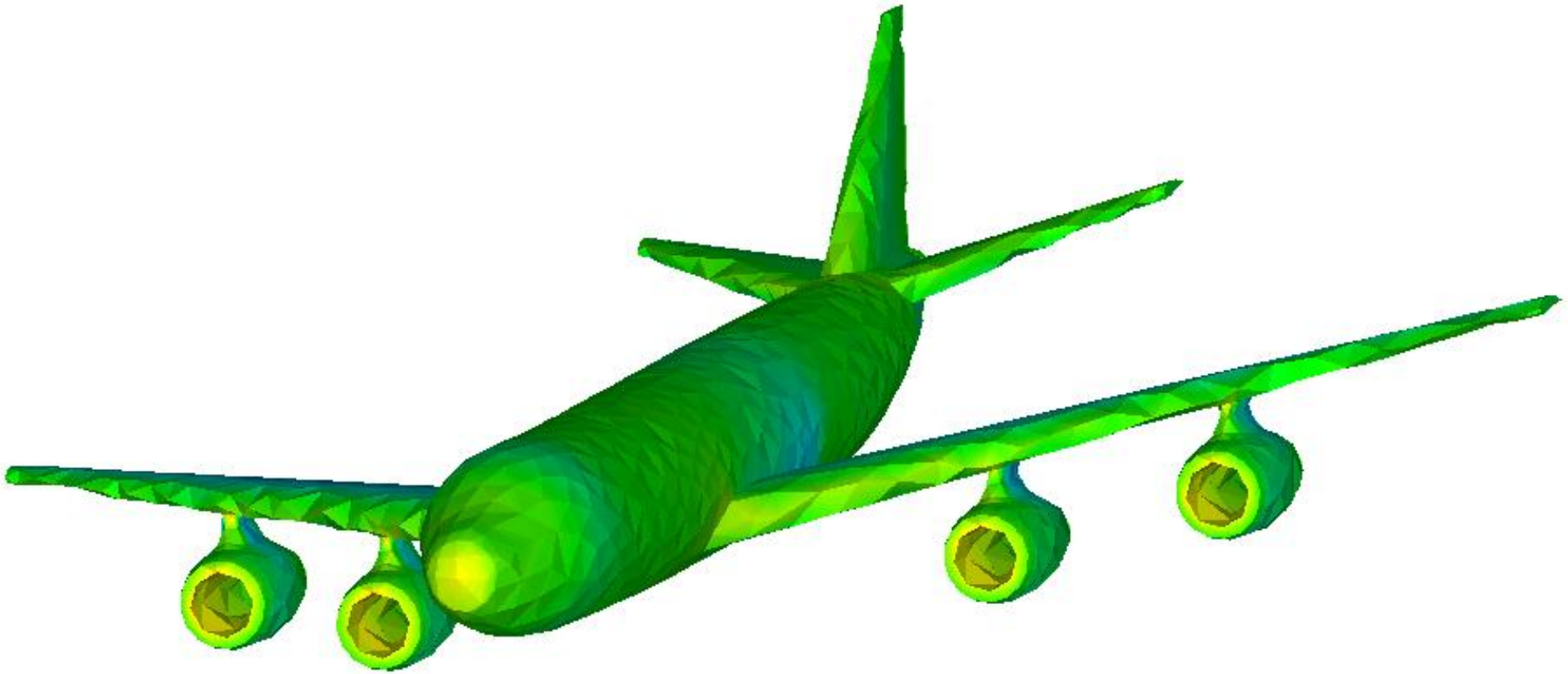






Тетраэдральные сетки

10^8 *узлов*



Литература

1. *Fiedler M.* A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory // Czechoslovak Mathematical Journal. – 25(100) – Praha, 1975. – Pp. 619–633.
URL: http://www3.math.tu-berlin.de/Vorlesungen/SS14/MatricesGraphsPDEs/paper_for_students/CZMathJ-25-1975-Fiedler%20copy.pdf
2. *B. Hendrickson, R. Leland*. A Multilevel Algorithm for Partitioning Graphs // Supercomputing '95 Proceedings. – San Diego, CA, 1995. http://www.leonidzhukov.net/hse/2016/networks/papers/MultilevelAlgorithmPartitioningGraphs.pdf
3. *George Karypis* Family of Graph and Hypergraph Partitioning Software URL: http://glaros.dtc.umn.edu/gkhome/views/metis/
4. *Alex Pothen, Horst D. Simon, and Kang-Pu Paul Liu.* Partitioning Sparse Matrices with Eigenvectors of Graphs. Report RNR-89-009, July 1989. URL: http://snap.stanford.edu/class/cs224w-readings/Pothen89Partition.pdf
5. *Е. Н. Головченко, Е. Ю. Дорофеева, В. А. Гасилов, М. В. Якобовский*. Вычислительный эксперимент по оценке качества алгоритмов параллельной декомпозиции больших сеток // *Препринты ИПМ им. М.В. Келдыша*. — 2013. — № 7. — С. 1–32. URL: http://keldysh.ru/papers/2013/prep2013_07.pdf
6. *М.А. Корнилина, М.В. Якобовский.* Оценка накладных расходов при выполнении расчетов на локально измельчаемых сетках.
// Препринты ИПМ им.М.В.Келдыша. - 2022. - № 102. - С. 1-36.
URL: https://library.keldysh.ru/preprint.asp?id=2022-102

Контакты

Якобовский М.В., *чл.-корр. РАН, проф., д.ф.-м.н.,*

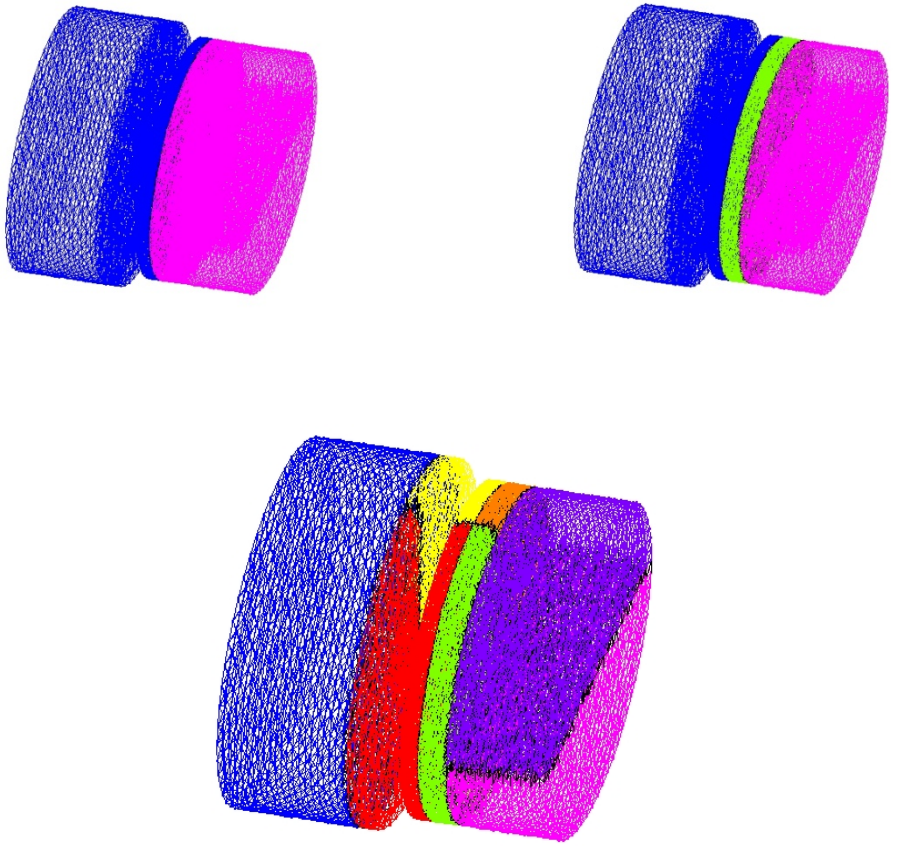
Заместитель директора по научной работе
Института прикладной математики
им. М.В.Келдыша Российской академии наук

mail: lira@imamod.ru

web: http://lira.imamod.ru

Параллельный алгоритм геометрической декомпозиции сеток

- **рекурсивная координатная бисекция**



используется распределенная сортировка

Разбиения на микродомены

дисбаланс числа вершин в 25600 микродоменах, %

Методы	Сетка 1	Сетка 2	Сетка 3	Сетка 4
разбиение графов				
IncrDecomp	3,5	0,1	0,3	0,2
PartKway	53,4	59,8	58,6	64,3
PartGeomKway	48,7	50,4	62,4	56,5
PT-Scotch	8,3	8,3	8,3	8,3
геометрические методы				
GeomDecomp	0,01	0,01	0,02	0,01
RCB	0,01	0,01	0,02	0,01

138

Разбиения на микродомены

число несвязных микродоменов из 25600

Методы	Сетка 1	Сетка 2	Сетка 3	Сетка 4
разбиение графов				
IncrDecomp	0	0	0	1
PartKway	69	35	37	29
PartGeomKway	67	34	28	37
PT-Scotch	7	0	2	4
геометрические методы				
GeomDecomp	62	38	16	33
RCB	64	43	14	44

139

Пакет MARPLE3D

(ИПМ им. М.В.Келдыша РАН)

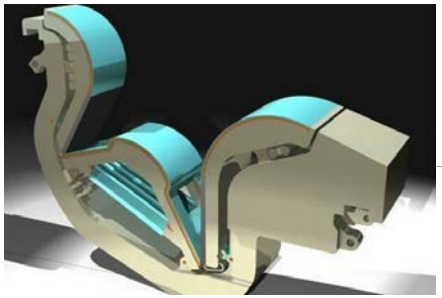
- предметная область – двухтемпературная радиационная магнитная гидродинамика

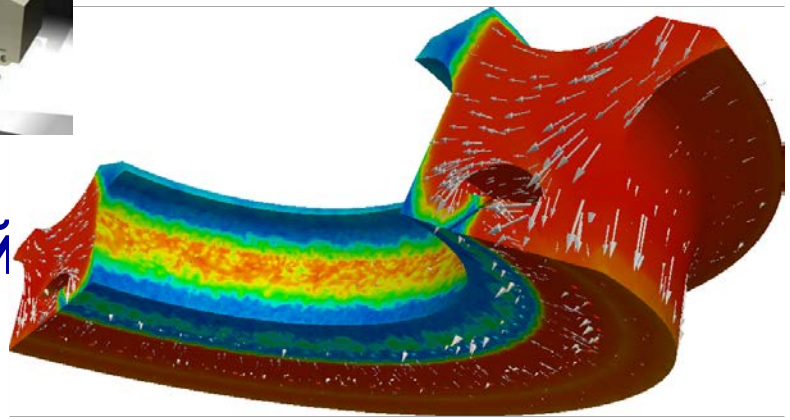
- **На задачах газовой динамики проведено тестирование разбиений, полученных методами пакетов GridSpiderPar, ParMETIS, Zoltan и PT-Scotch**
- **Сравнивалась эффективность параллельного счета физических задач пакетом MARPLE3D при распределении сеток по ядрам в соответствии с различными разбиениями**

140

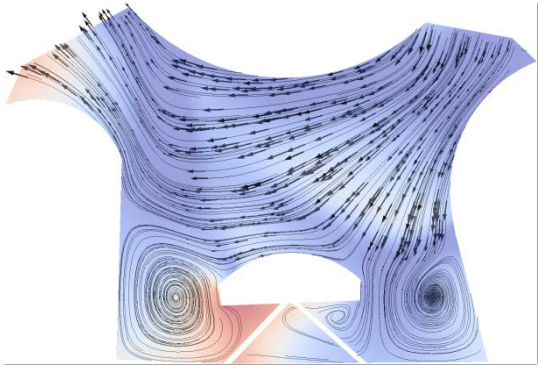
Моделирование газоплазменных потоков в диверторе токамака ITER





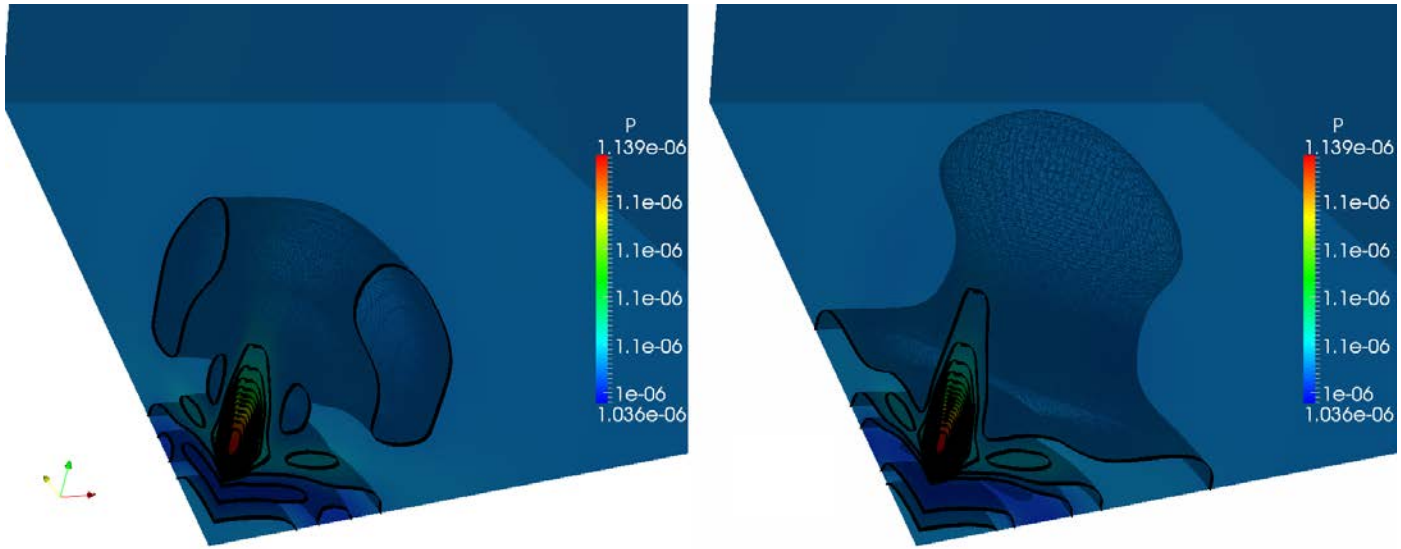


- **Полная система уравнений газовой динамики**
- **Турбулентная вязкость**
- **Радиационный и кондуктивный теплоперенос**
- **Явные и неявные схемы**



141

Моделирование распространения ударной волны от приземного источника энергии взрывного типа

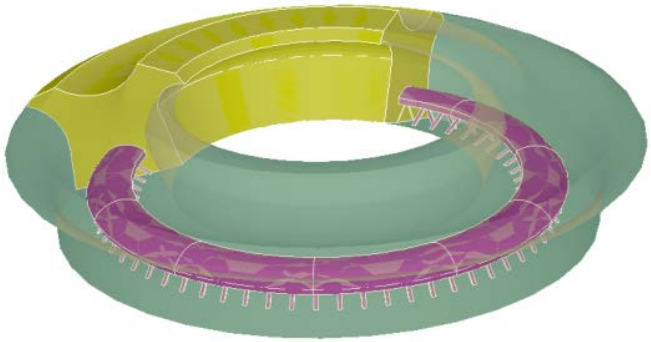


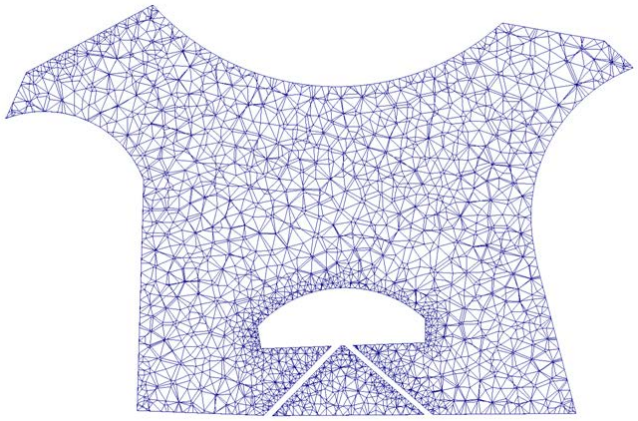
- **Полная система уравнений газовой динамики**
- **Кондуктивный теплоперенос**
- **Явные и неявные схемы**

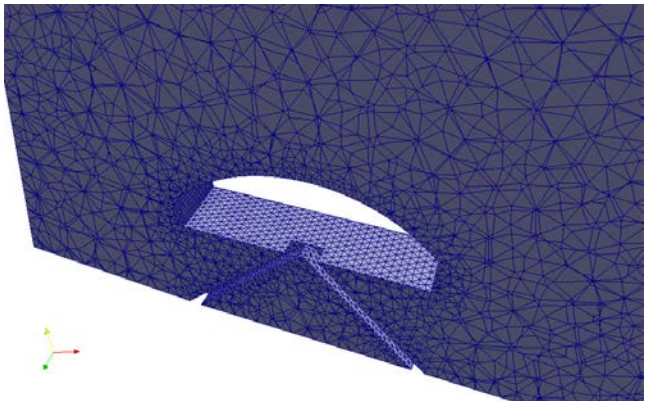
142

Расчетные сетки

Дивертор токамака (divertor)





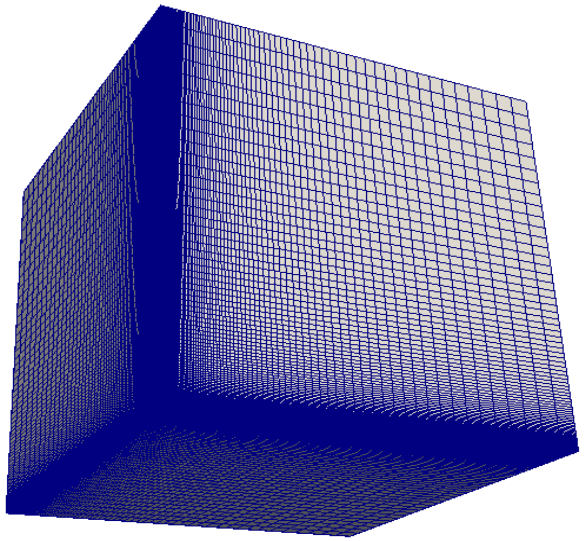


- Тетраэдральная сетка (3 миллиона ячеек)
- сгущение сетки вблизи мелких объектов
- 256 доменов

143

Расчетные сетки

Приземный взрыв (boom и boomL)



Гексаэдральные сетки
61 миллион ячеек для “boom”
116 миллионов ячеек для “boomL”

Параллелепипеды с различными aspect ratio

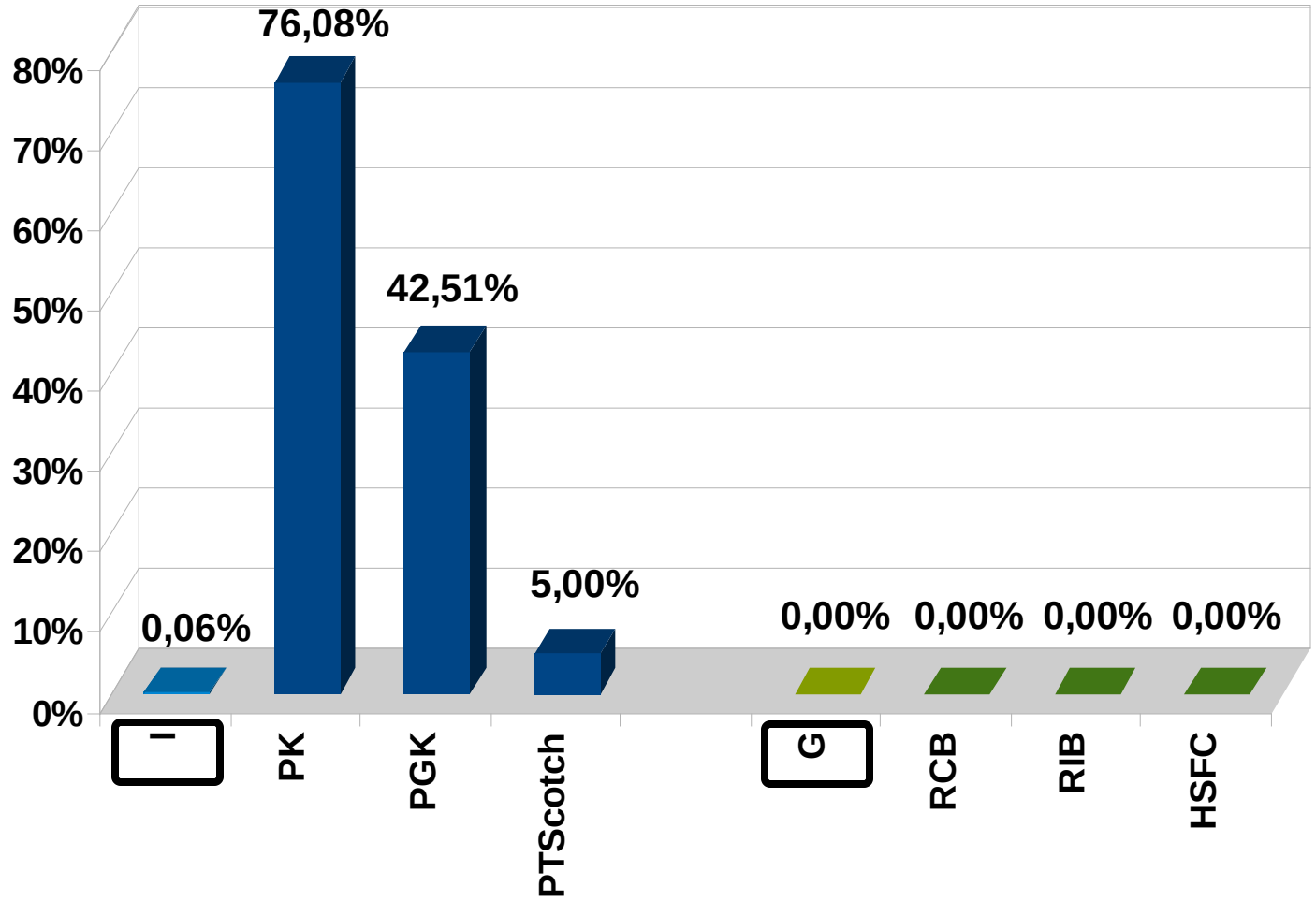
boom:
- 4096 доменов

boomL:
- 10080 доменов

- Для всех расчетных сеток были построены дуальные графы с числом вершин $2.8 \cdot 10^6$ - $1.2 \cdot 10^8$ и числом ребер $2.3 \cdot 10^7$ - $1.0 \cdot 10^9$
- Вычисления проводились на кластерах МВС-100К (227,94 TFlop/s), "Ломоносов" (1700 TFlops) и «Helios» (1524.1 TFlop/s)

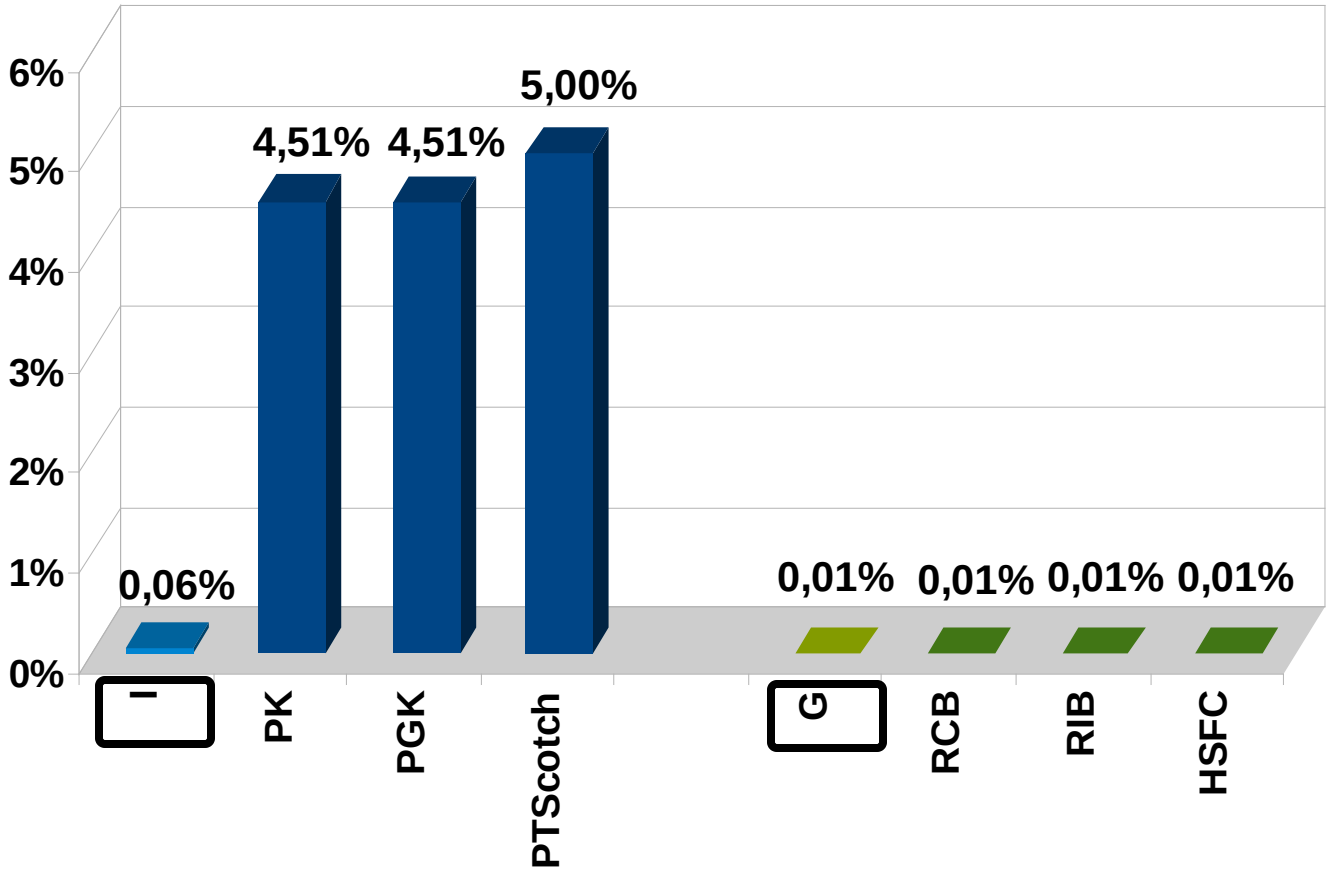
144

Дисбаланс числа вершин в доменах: недостаток вершин (boom)



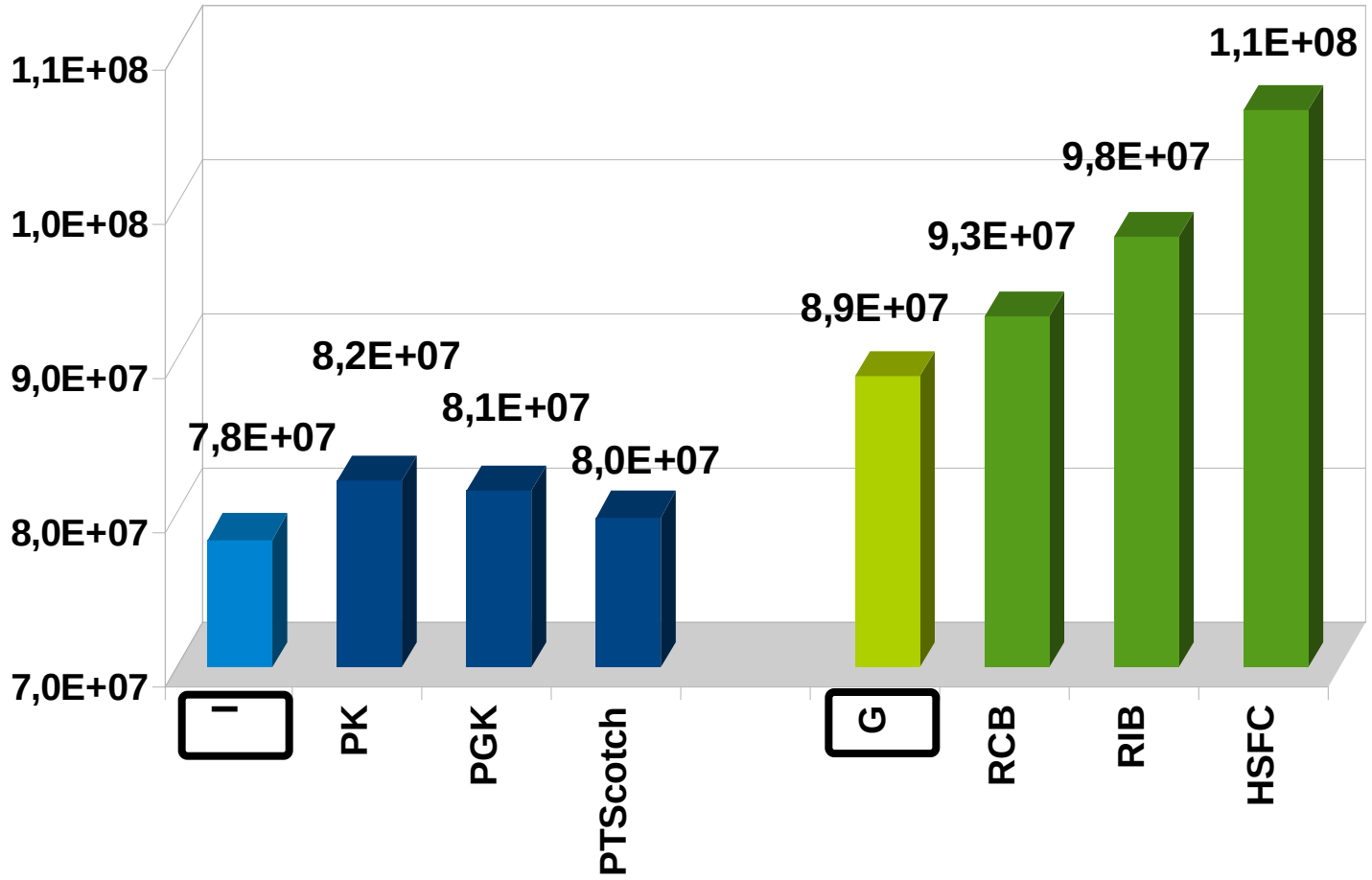
145

Дисбаланс числа вершин в доменах: избыток вершин (boom)



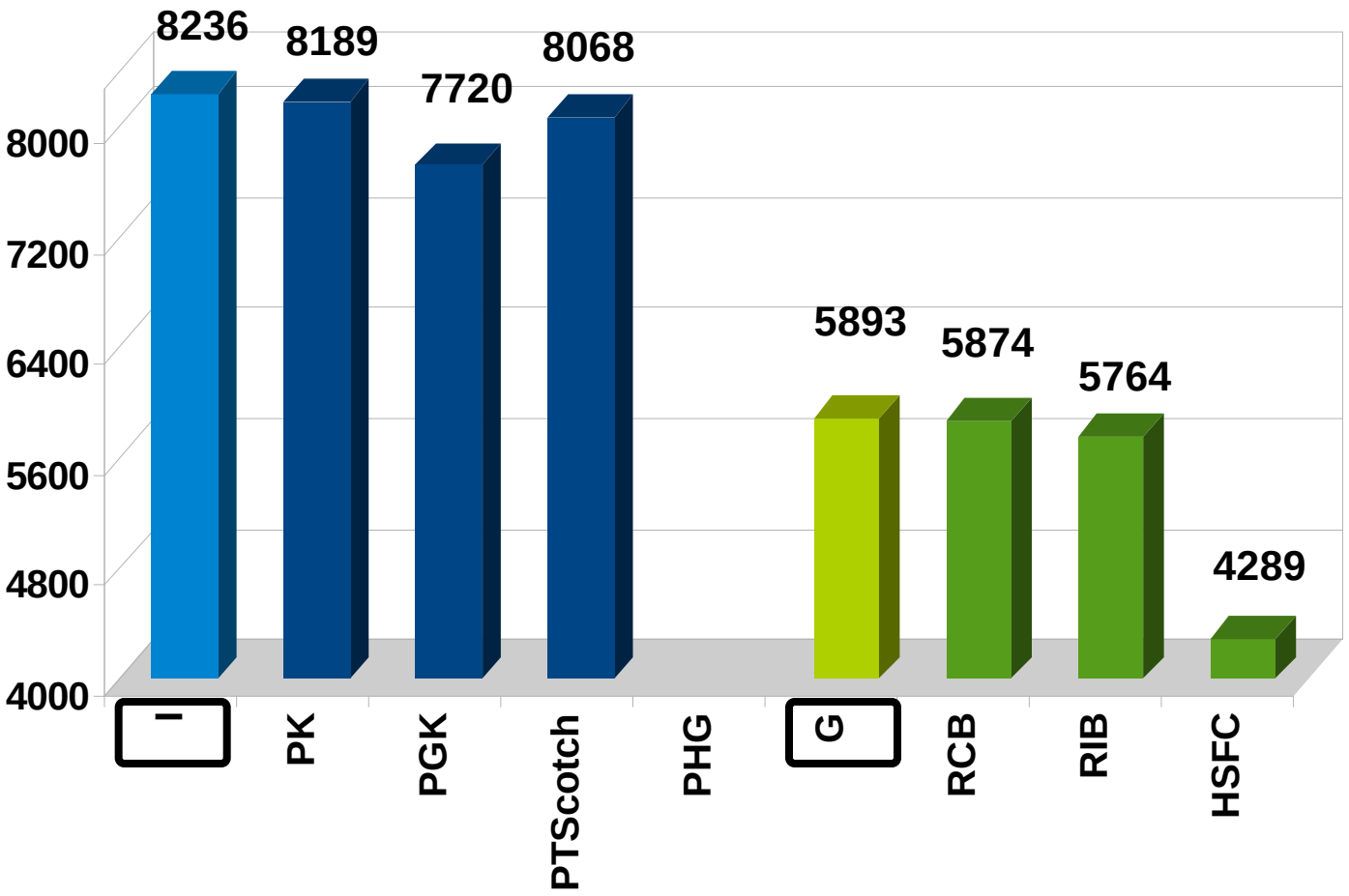
146

Число разрезанных ребер (boomL)



147

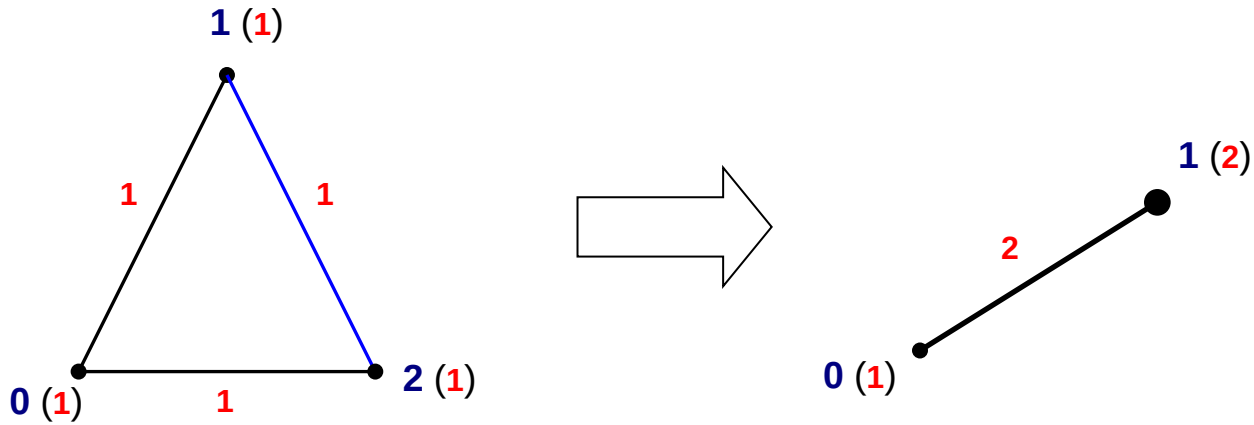
Число шагов по времени (divertor)

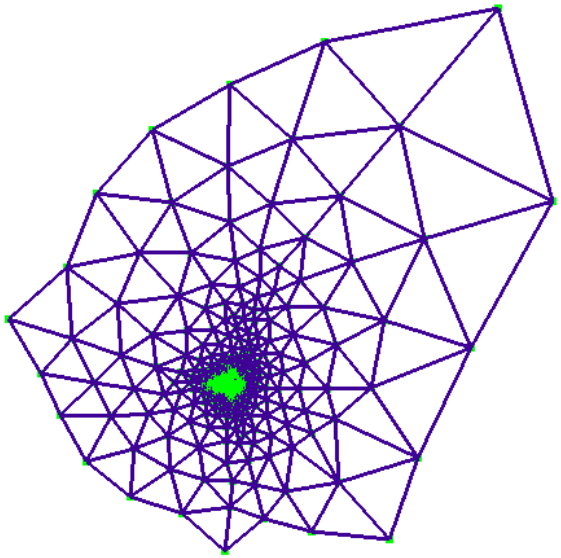


148

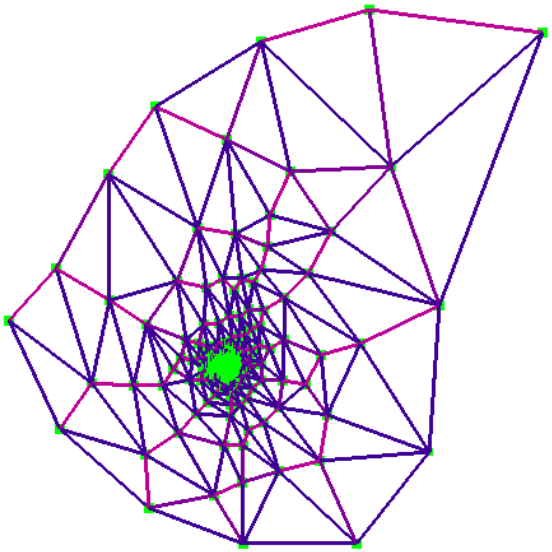
Огрубление на этапе предекомпозиции

огрубление

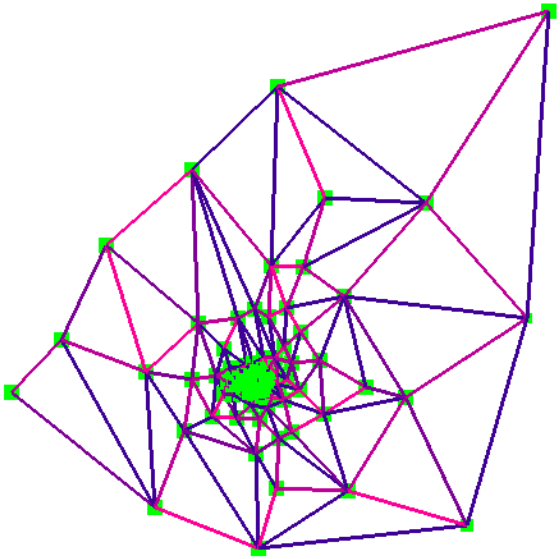




исходная сетка



1-ое огрубление



2-ое огрубление

150