



ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 128 за 2017 г.



ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

Суков С. А.

Метод сжатия топологии
гибридных сеток и
библиотека подпрограмм
hm4PackLib

Рекомендуемая форма библиографической ссылки: Суков С. А. Метод сжатия топологии гибридных сеток и библиотека подпрограмм hm4PackLib // Препринты ИПМ им. М.В.Келдыша. 2017. № 128. 15 с. doi:[10.20948/prepr-2017-128](https://doi.org/10.20948/prepr-2017-128)
URL: <http://library.keldysh.ru/preprint.asp?id=2017-128>

**Ордена Ленина
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Российской академии наук**

С.А. Суков

**Метод сжатия топологии
гибридных сеток
и библиотека подпрограмм
hm4PackLib**

Москва — 2017

Суков С.А.

Метод сжатия топологии гибридных сеток и библиотека подпрограмм hm4PackLib

В работе рассматривается проблема сжатия представления геометрии неструктурированных сеток. Приводится описание предлагаемого автором метода упаковки топологии сеток с элементами типа тетраэдр, треугольная призма, четырехугольная пирамида и гексаэдр, а также разработанной на его основе библиотеки подпрограмм hm4PackLib. Представлены результаты последовательного и распределенного сжатия гибридных сеток, содержащих более $5 \cdot 10^8$ элементов.

Ключевые слова: неструктурированные гибридные сетки, сжатие данных

Sergey Alexandrovich Sukov

A compression method for a topology of hybrid mesh and a subroutine library hm4PackLib

This paper contains the description of unstructured mesh compression method and the library hm4PackLib. The proposed method may be applied to obtain compact topology representation of hexahedron, tetrahedron, prism and pyramid meshes data. An author presents results of sequential and distributed data compression of meshes, which contains up to $5 \cdot 10^8$ elements.

Key words: unstructured meshes, data compression

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект № 15-07-04213-а, 16-07-00206-а.

Оглавление

1. Введение.....	3
2. Сжатие геометрии гибридных сеток	3
2.1. Формат представления данных.....	4
2.2. Сжатие координат	5
2.3. Метод сжатия топологии.....	5
2.4. Результаты вычислительных экспериментов	9
3. Распределенная обработка данных.....	11
4. Библиотека подпрограмм hm4PackLib.....	12
5. Заключение.....	14
6. Список литературы	15

1. Введение

Вычислительная мощность современных суперкомпьютерных комплексов позволяет решать задачи, сопряженные с обработкой больших объемов данных. Применительно к моделированию задач математической физики это, в частности, означает возможность проведения расчетов на подробных сетках. Большинство параллельных алгоритмов теоретически масштабируются на произвольное количество вычислителей. Однако эффективность их программных реализаций сильно обусловлена особенностями применяемых подходов к хранению и распределенной обработке данных.

Основным методом дискретизации областей сложной формы считаются неструктурированные сетки, которые могут состоять из многогранников различных типов. Представление геометрии гибридной сетки большой размерности занимает кратный гигабайтам объем дискового пространства. Поэтому для повышения эффективности копирования, хранения и обработки данных необходимо решать задачу их сжатия.

В настоящей работе приводится описание предлагаемого автором метода сжатия топологии сеток с элементами типа тетраэдр, треугольная призма, четырехугольная пирамида и гексаэдр, а также разработанных на его основе программных средств.

2. Сжатие геометрии гибридных сеток

Программы численного моделирования задач математической физики сеточными методами в подавляющем большинстве работают с дискретными моделями, состоящими из многогранных ячеек четырех типов: тетраэдр, треугольная призма, четырехугольная пирамида и гексаэдр. Описание геометрии гибридной сетки помимо координат узлов требует указания в явном виде топологии каждого сеточного элемента. Типы сеточных элементов и списки индексов их вершин формируют основной объем данных. Так, например, для тетраэдральной сетки с соотношением 1:5 между числом узлов и числом тетраэдров топологическое представление элементов займет более 75% от суммарного объема записываемых данных. Задачи сжатия массива координат и топологии ячеек имеют разную специфику и, как правило, решаются различными методами. Большинство предложенных в данной области подходов ориентированы на обработку триангуляции [1, 2]. Ниже приводится обобщенный формат записи неструктурированных сеток с элементами четырех типов, описание специального метода упаковки топологии многогранников и результатов его применения, а также формулировка алгоритма распределенной компрессии сеток большого размера.

2.1. Формат представления данных

Описание сетки должно содержать данные, которых достаточно для вычисления геометрических параметров элементов и определения отношений их взаимного расположения. Вне зависимости от типа рассматриваемой задачи и программного обеспечения формат представления геометрии в большинстве случаев, по сути, совпадает.

Внутри сетки, содержащей N_V узлов и N_E многогранников, вводится сквозная целочисленная индексация узлов ($V_{id} = 0, 1, 2, \dots, N_V - 1$) и элементов ($E_{id} = 0, 1, 2, \dots, N_E - 1$). Описание сетки делится на координатный и топологический блоки данных. Координатный блок состоит из N_V записей с индексом и тройкой пространственных координат сеточного узла. Топологический блок содержит N_E записей с индексом элемента, его типом и упорядоченным списком индексов образующих элемент вершин. В оперативной памяти эти данные можно структурировать в виде трех линейных массивов. Сгруппированные по вершинам тройки координат записываются в массив `double coord[Nv*3]` в порядке возрастания V_{id} . Топология сеточных элементов хранится в массивах `int XE[NE+1]` и `int AE[...]`. Смещение начала списка вершин элемента E_{id} записывается в ячейку `XE[Eid]`. Сам список занимает ячейки памяти `AE[XE[Eid]], ... AE[XE[Eid+1]-1]`. Для сеток, состоящих из тетраэдров, треугольных призм, четырехугольных пирамид и гексаэдров тип E_{id} однозначно определяется числом вершин ($XE[Eid+1] - XE[Eid] = 4, 5, 6, 8$). Идентичный формат представления данных используется, например, в подпрограммах библиотеки ParMetis [3].

Для корректной обработки элементов необходима предварительная фиксация алгоритма перечисления вершин многогранников. В данной работе предполагается, что упорядочивание индексов вершин (рис. 1) соответствует формату представления данных генератора GAMBIT [4].

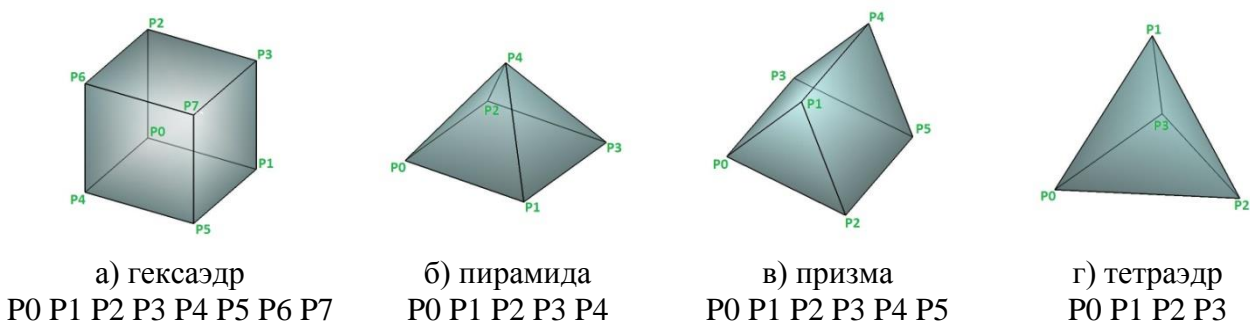


Рис. 1. Алгоритм перечисления индексов вершин элементов

При проведении численных расчетов описание сетки часто дополняется блоком данных с признаками элементов, границ расчетной области и т.д. Формат их записи выбирается разработчиком программного обеспечения.

Поэтому наиболее универсальным подходом к упаковке данных блока будет применение метода сжатия без потерь.

2.2. Сжатие координат

Специальные методы компрессии координат, которые обеспечивают высокий коэффициент сжатия, относятся к классу методов кодирования с потерей точности. А следовательно, существует вероятность нарушения (вырождения, выворачивания) элементов после округления значений. Поэтому применение этих методов с целью упаковки координат узлов расчетных сеток в общем случае невозможно.

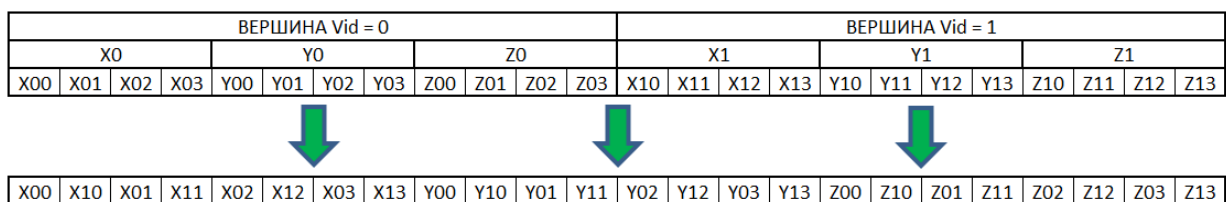


Рис. 2. Пример сортировки байт массива вещественных чисел (тип float 4 байта) с координатами двух вершин

Для сжатия данных координатного блока чаще используют алгоритмы компрессии без потерь и соответствующие библиотеки подпрограмм. К их числу относится, например, свободно распространяемая кроссплатформенная библиотека zlib [5]. В некоторых случаях возможно повышение эффективности сжатия координат путем предварительной сортировки байт массива coord. Данные группируются по осям X, Y, Z, а внутри группы сортируются по байтам (рис. 2).

2.3. Метод сжатия топологии

Алгоритмы сжатия без потерь гарантируют тождественность исходных данных и результата их упаковки с последующим восстановлением. Применительно к топологическому блоку такая жесткая постановка задачи не является обязательной. Метод упаковки должен обеспечить только возможность корректной инициализации топологии многогранников по восстановленным данным. Поэтому для достижения максимального коэффициента сжатия допустимы изменения в порядке перечисления вершин многогранников и порядке перечисления списков вершин элементов в массиве AE с соответствующей корректировкой массива XE. Изменение массива индексов вершин многогранника с соблюдением соглашения о порядке перечисления вершин и отличное от исходного упорядочивание сеточных элементов, очевидно, не влияют на корректность дискретизации объема расчетной области. Однако переход к сжатому представлению топологии

требует модификации блока данных с дополнительными признаками сеточных примитивов, если такой блок данных существует.

Предлагаемый метод сжатия топологии гибридных сеток основан на принципах обработки триангуляции методом полос [6]. Исходное множество многогранников разбивается на последовательности или цепочки элементов с общей гранью. Топология начала цепочки записывается в неупакованном виде. Каждый последующий элемент задается своим типом, номером общей грани предшествующего элемента цепочки и упорядоченным списком недостающих узлов.

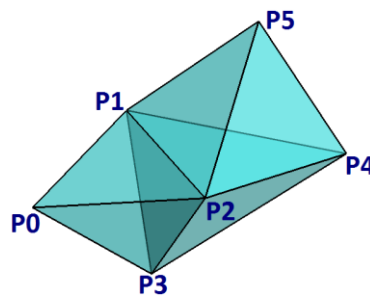


Рис. 3. Пример цепочки тетраэдров

На рис. 3 показан пример цепочки, состоящей из тетраэдров $P_0P_1P_2P_3$, $P_1P_2P_3P_4$ и $P_1P_2P_4P_5$. Первый тетраэдр описывается четверкой своих вершин. Второй тетраэдр строится на третьей грани (индексация граней в соответствии с [7]) первого тетраэдра с добавлением вершины P_4 . Тетраэдр $P_1P_2P_4P_5$ содержит вторую грань предыдущего элемента цепочки и вершину P_5 .

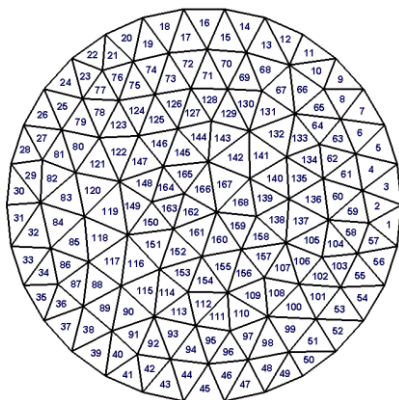
В сжатом формате для хранения номеров граней отводится 2 (тетраэдр) или 3 (гексаэдр, пирамида, призма) бита, а тип любого многогранника задается 2 битами. Для реконструкции пирамиды на треугольной грани в сжатое описание элемента добавляется пара дополнительных бит. Они помечают одну из вершин треугольника как вершину пирамиды. В случае, когда призма строится на четырехугольной грани, один дополнительный бит задает пару противоположных ребер четырехугольника, принадлежащих треугольным граням призмы.

Таблица 1

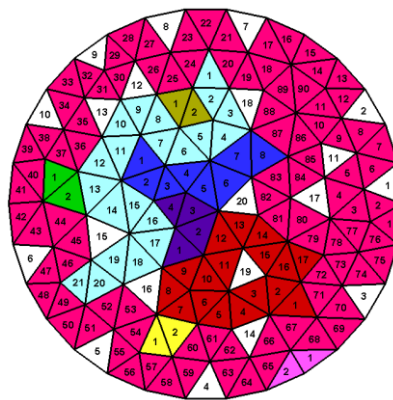
Элемент	Базовая грань	Число служебных бит	Число дополнительных вершин	Коэффициент сжатия
Тетраэдр	Треугольник	3	1	3.66
Пирамида	Треугольник	5	2	2.32
	Четырехугольник	3	1	4.57
Призма	Треугольник	3	3	1.94
	Четырехугольник	4	2	2.82
Гексаэдр	Четырехугольник	3	4	1.95

В табл. 1 даны коэффициенты упаковки отдельных многогранников. Представленные значения учитывают только сжатие списков вершин без идентификатора типа. В исходном формате идентификатор типа неявно содержится в массиве ХЕ. То есть можно считать, что для хранения типа элемента выделяется одна целочисленная переменная (4 байта). Таким образом, переход к представлению типа двумя битами дает равный 16 коэффициент сжатия для массива ХЕ, размер которого составляет от 11% до 20% от общего объема данных.

В соответствии с табл. 1 коэффициент сжатия принимает значения от 1.94 до 4.57. Наихудший показатель соответствует реконструкции шестигранных элементов и призм от треугольных граней, что объясняется большой относительной длиной списков недостающих вершин. Наиболее эффективный метод их компрессии должен учитывать особенности подходов к генерации сеток. Оптимальный метод упаковки здесь будет состоять в выделении структурированных блоков и полученных протягиванием подобластей сетки с их последующей записью в минимальном по размеру формате. Максимальный коэффициент сжатия относится к варианту построения пирамиды на четырехугольной грани. Однако доля пирамид обычно не превышает нескольких процентов от общего числа сеточных элементов. И, таким образом, можно прогнозировать, что на практике осредненное значение коэффициента упаковки топологического блока гибридной сетки будет находиться в пределах 2.5 раз.



а) цепочка, проходящая по всем ячейкам



б) построение нескольких цепочек внутри сетки

Рис. 4. Примеры построения цепочек смежных элементов на треугольной сетке

Эффективность обсуждаемого метода сжатия зависит от вида решения задачи, состоящей в выделении цепочек смежных по грани сеточных элементов или, другими словами, задачи построения цикла по вершинам дуального графа. Оптимальным результатом здесь, очевидно, будет цепочка, проходящая по всем вершинам графа (рис. 4а). Эта задача относится к классу NP-полных. Поэтому для ее решения предложен быстрый алгоритм, в результате работы которого

внутри графа в общем случае выделяется несколько последовательностей (рис. 4б).

Алгоритм построения цепочек анализирует дуальный сеточный граф. Вершинами графа являются сеточные многогранники, а ребрами – их связи по смежным граням. В начале процедуры рассматривается граф, содержащий все сеточные ячейки. В процессе работы алгоритма вершины графа, добавленные к одной из цепочек, исключаются из рассмотрения. Начало цепочки выбирается произвольным образом. Далее в нее добавляется одна из вершин графа, связанная ребром с предыдущей вершиной, и т.д. Рассматривались три варианта продолжения цепочки:

- добавление вершины случайным образом (вариант 1);
- добавление вершины с максимальным числом ребер (вариант 2);
- добавление вершины с минимальным числом ребер (вариант 3).

В табл. 2 приводятся параметры цепочек, построенных внутри тетраэдральной сетки. Сетка содержит 444635 узлов, 2369186 тетраэдров и 4638445 внутренних граней, соответствующих ребрам дуального графа. Результаты тестирования разных версий алгоритма представлены для двух вариантов индексации сеточных элементов. К числу неупакованных элементов относятся начала цепочек, а также "выколотые" многогранники или, другими словами, цепочки нулевой длины (множество треугольников без заливки в примере на рис. 4б). При вычислении осредненной длины цепочек учитываются только последовательности длиной более одного элемента.

Таблица 2

Индексация	Алгоритм добавления вершин	Число и доля неупакованных элементов	Число цепочек	Максимальная длина цепочки	Средняя длина цепочки
1	Вариант 1	180113 (7.6%)	138926	778	16
	Вариант 2	399199 (16.9%)	17134	1538	12
	Вариант 3	47004 (2.0%)	45745	2642	51
2	Вариант 1	101641 (4.3%)	90462	941	26
	Вариант 2	404065 (17.1%)	163805	1689	12
	Вариант 3	41805 (1.8%)	41005	2297	57

Результат работы алгоритма с добавлением вершин случайным образом сильно зависит от индексации сеточных ячеек. В рассматриваемом примере после изменения упорядочивания многогранников число неупакованных элементов сокращается в 1.77 раза, а средняя длина цепочки возрастает в 1.62 раза. Однако в обоих случаях алгоритм добавления вершин случайным образом оказывается эффективнее менее чувствительного к индексации алгоритма выбора вершины с наибольшим числом вариантов продолжения цепочки. Акцент на генерации цепочек максимальной длины приводит к увеличению доли выколотых элементов. Наилучшие показатели демонстрирует третья

версия алгоритма, где продолжением цепочки становится вершина графа с минимальным числом ребер.

2.4. Результаты вычислительных экспериментов

В данном разделе представлены оценки эффективности различных подходов к упаковке данных, полученные при обработке четырех гибридных сеток. Размерности и состав сеток даны в табл. 3.

Таблица 3

Параметр		Сетка			
		МА	МВ	МС	МД
Число узлов		1030301	190542	159750	50357384
Число элементов	Тетраэдров	0	1083344	440160	16160888
	Пирамид	0	0	19000	276516
	Призм	0	0	64200	0
	Гексаэдров	1000000	0	40000	47409728
	Всего	1000000	1083344	563360	63847132

Сжатие координатной информации выполнялось с использованием подпрограмм библиотеки `zlib`. Коэффициенты упаковки массива координат узлов в исходном виде и после сортировки байт представлены в табл. 4. Переупорядочивание байт заметно улучшает коэффициент сжатия только для сетки МА, которая представляет собой кубическую решетку $100 \times 100 \times 100$ элементов. В остальных случаях эффект от переупорядочивания не превышает 20% и при этом может быть как положительным, так и отрицательным. Поэтому оптимальным подходом к упаковке координатного блока выглядит процедура, предусматривающая возможность выбора наилучшего из двух вариантов представления исходных данных.

Таблица 4

Сетка	Коэффициент сжатия координат (<code>zlib</code>)	
	Без сортировки	С предварительной сортировкой
МА	9.02	116.43
МВ	1.40	1.49
МС	1.82	2.13
МД	1.54	1.33

В рамках задачи компрессии топологического блока сравнивались три подхода. Во-первых, сжатие библиотекой `zlib`. Во-вторых, сжатие средствами библиотеки подпрограмм `hm4PackLib`, разработанной на основе предложенного метода упаковки сеток. И, в-третьих, комбинация первого и второго подходов,

когда полученное при помощи hm4PackLib сжатое представление затем упаковывается средствами zlib.

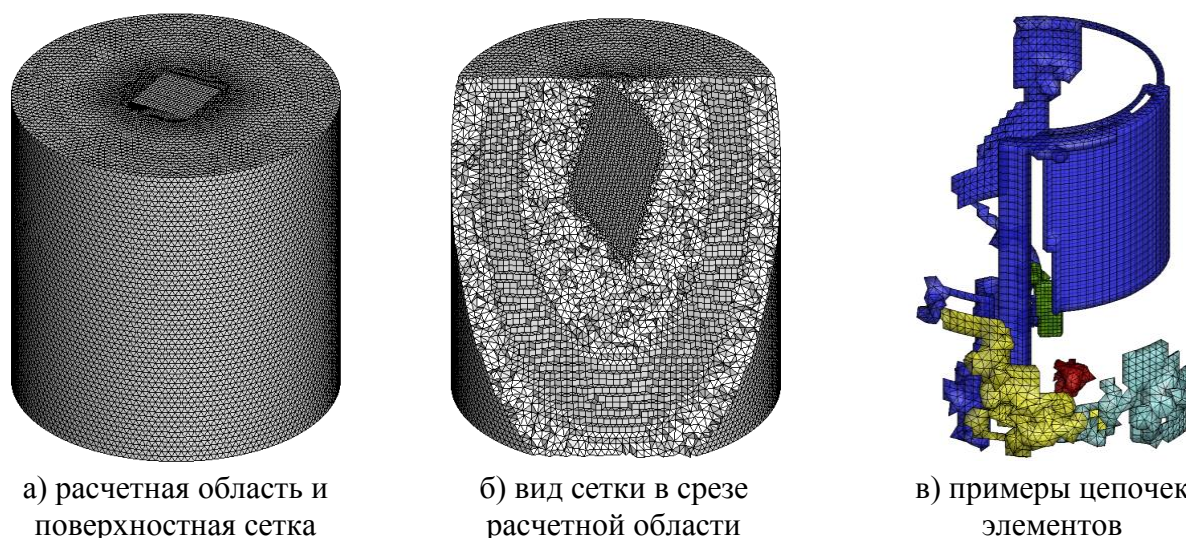


Рис. 5. Гибридная сетка MC

На рис. 5 показана расчетная область для сетки MC (рис. 5а), срез сетки в сечении расчетной области (рис. 5б) и примеры нескольких цепочек (рис. 5в), построенных во время сжатия топологии средствами библиотеки hm4PackLib. Коэффициенты упаковки приводятся в табл. 5.

Таблица 5

Сетка	Коэффициент сжатия топологии		
	zlib	hm4PackLib	Комбинация
MA	2.36	2.17	4.86
MB	2.36	4.17	7.95
MC	2.24	3.31	6.77
MD	3.25	2.30	4.79

Из представленных результатов следует, что использование комбинации двух методов позволяет сократить объем записываемых на диск данных в 1.5 и более раз по сравнению со сжатием топологии библиотекой zlib. Максимальный показатель упаковки данных достигается при работе с преимущественно тетраэдральными сетками MB и MC, где hm4PackLib демонстрирует наилучший результат. Снижение показателя компрессии для сеток MA и MD объясняется большим относительным числом гексаэдральных элементов в их составе.

В табл. 6 приводятся характерные времена различных этапов упаковки и разархивирования сетки MD. Эксперимент проводился на стационарном ПК с процессором Intel Core i7 6700 K 4.0 GHz и 64 Гб оперативной памяти.

Таблица 6

Этап		Время (секунд)		
hm4PackLib	Инициализация дуального графа	71.5	93.3	128.1
	Инициализация цепочек	5.8		
	Упаковка	16.0		
Упаковка zlib		34.8		
Декомпрессия zlib		3.6	9.0	
Декомпрессия hm4Pack		5.4		

3. Распределенная обработка данных

Эффективная генерация и обработка неструктурированных сеток, содержащих более $5 \cdot 10^7$ элементов, возможна только в параллельном режиме. Для распределенного сжатия сетку необходимо разделить на части, а затем независимо упаковать каждую из подобластей. При использовании подпрограмм hm4PackLib декомпозиция должна соответствовать критериям метода геометрического параллелизма. Тогда сетка будет разделена на связанные подобласти одинакового размера (рис. 6), что гарантирует возможность построения цепочек. Массив координат разбивается на части простым делением интервала индексов вершин на равные отрезки.

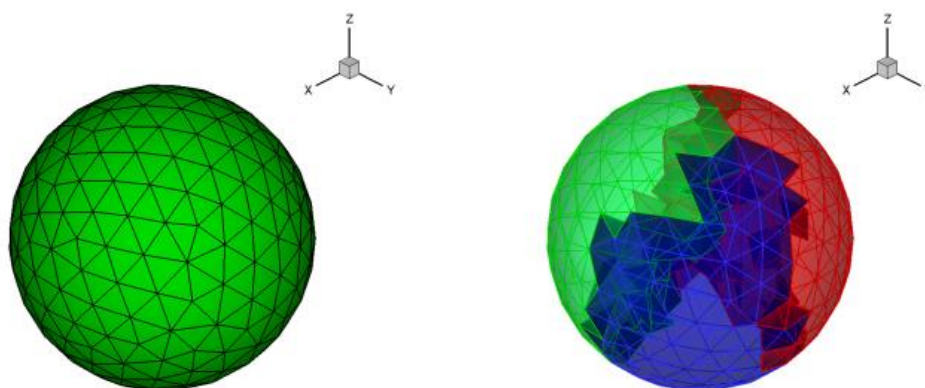


Рис. 6. Пример декомпозиции расчетной сетки на три связанные подобласти

После декомпозиции представление сетки трансформируется к множеству блоков сжатых данных, которые затем записываются в один или несколько файлов. Последний вариант более предпочтителен с точки зрения копирования данных между удаленными вычислительными системами.

Оценка эффективности распределенной обработки данных проводилась на примерах сжатия тетраэдральной и гибридной сеток большого размера. Первая сетка содержит 87 151 790 узлов и 521 303 552 тетраэдра (11.7 ГБ). В состав второй сетки входит 78 221 097 узлов, 236 609 920 тетраэдров, 4 104 000

пирамид, 32 870 400 призм и 20 480 000 гексаэдров (7.8 ГБ). Распределенная декомпозиция сеток выполнялась средствами библиотеки GrdSpiderPar [8], для выделения геометрии подобластей использовался функционал библиотеки hm4PreprocessorLib [9]. Полученные результаты компрессии данных приводятся в табл. 7.

Таблица 7

		Тетраэдральная сетка			Гибридная сетка		
		48	72	96	48	72	96
Координатный блок данных (zlib)							
Коэффициент сжатия	Минимум	1.11	1.08	1.08	1.13	1.12	1.09
	Максимум	1.19	1.21	1.21	5.74	5.83	6.49
	Общий	1.14	1.14	1.14	1.52	1.52	1.52
Топологический блок данных (hm4PackLib + zlib)							
Коэффициент сжатия	Минимум	9.30	9.30	9.30	5.94	5.22	4.99
	Максимум	9.39	9.40	9.40	8.54	8.93	9.00
	Общий	9.33	9.33	9.33	7.31	7.32	7.32

Декомпозиция координатного и топологического блоков выполнялась на одинаковое число частей. Оценивался как общий коэффициент сжатия данных, так и его минимальное и максимальное значение по отдельным подобластям. Коэффициент сжатия топологического блока соответствует последовательному сжатию hm4Pack и zlib. Компрессия данных координатного блока алгоритмом zlib выполнялась в исходном виде без сортировки байт.

В соответствии с представленными в табл. 7 значениями параметров осредненный коэффициент сжатия слабо зависит от степени измельчения, а его значение близко к результатам последовательной упаковки сеток. Разница между значениями локальных коэффициентов сжатия блоков гибридной сетки вполне ожидаема и объясняется различием топологической структуры блоков.

4. Библиотека подпрограмм hm4PackLib

Библиотека hm4PackLib содержит программную реализацию процедур сжатия и разархивирования топологии гибридных сеток на основе метода упаковки с выделением цепочек смежных элементов. Список доступных на уровне конечного пользователя функций библиотеки с описанием их назначения приводится в табл. 8.

Таблица 8

Подпрограмма	Назначение
hm4TopologyPack	Подпрограмма упаковки топологического блока данных сетки, состоящей из тетраэдров, треугольных призм, четырехугольных пирамид и гексаэдров
hm4TopologyUnPack	Подпрограмма разархивирования записанной в сжатом формате топологии гибридной сетки
dualGraphElementsSequence	Подпрограмма построения цепочек по элементам гибридной сетки
hm4CompareMeshes	Подпрограмма сравнения топологии двух сеток

Исходный текст программных модулей написан на языке C. Внутреннее распределение памяти осуществляется функцией `malloc`. В случае успешного завершения все подпрограммы возвращают значение 0. При возникновении ошибок в ходе их работы происходит аварийный выход из приложения с выводом в поток `stderr` информационного сообщения, содержащего описание ошибки. Далее приводятся прототипы библиотечных функций, выполняющих компрессию и разархивирование топологического блока данных.

```
int hm4TopologyPack(int NE, int *XE, int *AE, int sequenceTYPE,
                   int *N, void **TPSB,
                   FILE *stream);
```

Описание	Подпрограмма упаковки топологии гибридной сетки.
Параметры	
NE	Общее число сеточных элементов.
XE, AE	Указатели на массивы с топологическим описанием сеточных элементов.
sequenceTYPE	Алгоритм выделения цепочек элементов: 0 – добавление элемента случайным образом; 1 – добавление элемента с максимальным числом вариантов продолжения цепочки; 2 – добавление элемента с минимальным числом вариантов продолжения цепочки.
N	Указатель на переменную для записи длины массива (байт) с упакованным представлением топологии.
TPSB	Адрес переменной, в которую записывается указатель на массив данных с упакованным представлением топологии.
stream	Поток вывода консольных сообщений, отображающих статус работы подпрограммы. В случае <code>stream = NULL</code>

информационные сообщения не выводятся.

```
int hm4TopologyUnPack(int *NE, int **XE, int **AE,
                    int N, void *TPSB,
                    FILE *stream);
```

Описание Подпрограмма разархивирования сжатого представления топологии гибридной сетки.

Параметры

NE	Указатель на переменную для записи общего числа сеточных элементов.
XE, AE	Адреса переменных, в которые записываются указатели на массивы с топологическим описанием сеточных элементов.
N	Длина массива (байт) с упакованным представлением топологии.
TPSB	Указатель на массив данных с упакованным представлением топологии.
stream	Поток вывода консольных сообщений, отображающих статус работы подпрограммы. В случае stream = NULL информационные сообщения не выводятся.

5. Заключение

Метод сжатия топологии гибридных сеток, основанный на выделении последовательностей элементов, по эффективности отдельно сопоставим с методами упаковки данных без потерь. А применение их комбинации позволяет заметно сократить объем записываемых на диск данных. При этом описанный метод сравнительно просто адаптируется на случай распределенной обработки больших объемов данных. В то же время обобщенный подход к сжатию сеточных ячеек различных типов препятствует достижению максимальной эффективности компрессии гибридных сеток, содержащих структурированные блоки. Поэтому в дальнейшем предполагается совершенствование предложенного подхода путем добавления специальных алгоритмов выделения и обработки структурированных сеточных блоков.

6. Список литературы

1. Скворцов А.В., Костюк Ю.Л. Сжатие координат узлов триангуляции // Изв. вузов. Физика, 2002, № 5, с. 26-30.
2. Скворцов А. В. Сжатие топологических связей триангуляции // Выч. мет. программирование, 3:1 (2002), С. 124–132.
3. Электронный ресурс: ParMETIS.
<http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview>.
4. Электронный ресурс: GAMBIT. <http://www.ansys.com>
5. Электронный ресурс: zlib. <http://zlib.net/>
6. Evans F., Skiena S., Varshney A. Optimizing triangle strips for fast rendering // Proc. IEEE Visualization. 1996. P. 319–326.
7. Электронный ресурс:
https://web.stanford.edu/class/me469b/handouts/gambit_write.pdf.
8. Головченко Е.Н., Якобовский М.В. Пакет параллельной декомпозиции больших сеток GridSpiderPar // Вычислительные методы и программирование, М.: МГУ. — 2015. — Т. 16, № 4. — С. 507–517.
9. Суков С.А. Библиотека препроцессорной обработки неструктурированных сеток hm4PreprocessorLib // Препринты ИПМ им. М.В.Келдыша. 2016. № 108. 20 с. doi:10.20948/prepr-2016-108.