

Форматы данных для записи сеток

1 Введение

Современные инструменты решения задач вычислительной математики немыслимы без чтения из файлов начальных условий и записи результатов вычислений. На текущий момент существует огромное кол-во различных форматов для записи тех или иных расчётных данных: структурированные и не структурированные, однородные и не однородные сетки, сетки с данными в точках или в ячейках, многоблочные и иерархические сетки, с данными в виде значений, цветов, нормалей, потоков и т.д.. Под разные задачи подходят разные виды форматов.

Данная работы пытается сделать, по возможности, полный обзор тех форматов, с которыми мне довелось столкнуться. К сожалению, некоторые форматы проприетарны и работы с ними возможна только через специализированный софт, для некоторых требуется подключение сложной математики и геометрии (сплайновые и иерархические), такие форматы будут указаны только в скользь.

Для начала стоит ввести общие понятия и терминологию. Как правило задачи решаются в двумерном или трехмерном пространстве, но большинство форматов хранят трехмерные координаты, поэтому, если не указано иное, будем считать что мы работаем в трехмерном пространстве.

- Точка — точка в R^3 .
- Ячейка — точка, ребро (отрезок), многоугольник, многогранник.
- Грань ячейки — для многоугольника - ребро/отрезок, для многогранника - грань/многоугольник, для ребра - конец/точка, для точки не определено.
- Сетка — набор ячеек, точки сетки - вершины ячеек.

Как правило в коде сетка задается следующими данными:

- кол-во точек в сетке NP;
- массив точек P, массив размерности [NP][3] или [NP*3];

- кол-во ячеек;
- ячейки, каждая ячейка задается типом, кол-вом вершин в ячейке и массивом индексов вершин в массиве точек.

Важно отметить, что некоторые типы ячеек могут иметь различную нумерацию вершин, поэтому нужно заранее договориться о "правильной" нумерации. Например для гексаэдров распространены два типа нумерации.

Картинка с блоком и гексом.

Так же стоит обратить внимание на то, что индексы в массивах точек могут начинаться с 0 и с 1. Далее в работе считается что индексы нумируются с 0 до NP-1.

Для задания начальных условий в задачах аэрогидродинамики требуется определения различных граничных условий на разных частях общей границы тела, для этого к данным добавляются списки граней ячеек разбитые на группы, например, грани входящего/уходящего потока, грани обтекаемого тела. Наиболее удобный способ записи граничных условий таков:

- кол-во различных граничных условий NB
- для каждого граничного условия:
 - кол-во граней для данного граничного условия
 - список граней для данного граничного условия в одном из представлений
 - * грань как ячейка, те для тетраэдральной сетки - набор треугольников
 - * пара индексов — номер ячейки и локальный номер грани ячейки

Последний вариант предпочтительней, так как в данной структуре явно указывается связь между ячейками сетки и граничными условиями, в то время как в первом — данную связь необходимо восстанавливать.

Для вывода результатов решения необходимо записать данных расчетов вместе с сеткой. Для начала определим с какими данными мы будем работать.

- материал — целочисленное значение, как правило не отрицательное. Может под собой нести информацию об индексе, маске, цвете в палитре цветов и тд.

- функция — вещественное значение. Обычно используется для записи результатов расчета безразмерных величин: плотность, давление, температура; или характеристик сетки: качество, объем,
- текстурные координаты — два вещественных числа от 0.0 до 1.0. Используются для привязки текстуры к поверхности. Как правило в научных задачах не используется.
- цвет — три или четыре целых числа от 0 до 255 или вещественных числа от 0.0 до 1.0. Используется для задания цвета. Первые три числа задают координаты в RGB пространстве цветов, опциональное четвертое число прозрачность.
- вектор — три вещественных числа. Используется для хранения таких величин как силы, скорости, потоки и т.д..

Данные можно привязать к точкам, ячейкам и граничным условиям, причем в любом кол-ве и наборе. Однако в большинстве форматах можно записать только один набор данных. Это нужно учитывать.

2 Формат OFF

Формат OFF (Object File Format) - ASCII формат для представления полигональной поверхности. Есть бинарный вариант (?).

В этом формате можно хранить только многоугольные ячейки и ограниченный набор данных, но из-за простоты формата им бывает удобно пользоваться. Нумерация вершин начинается с 0.

Wolfram (<http://reference.wolfram.com/mathematica/ref/format/OFF.html>):

OFF 3D geometry format. Used for storing and exchanging 3D models. OFF is an acronym for Object File Format. Occasionally called COFF if color information is present. Related to NOFF and CNOFF. ASCII or binary format. Represents a single 2D or 3D object. Stores a collection of planar polygons with possibly shared vertices. Supports polygon and vertex colors and opacity specifications.

Geomview (<http://www.geomview.org/docs/html/OFF.html>):

OFF files (name for "object file format") represent collections of planar polygons with possibly shared vertices, a convenient way to describe polyhedra. The polygons may be concave but there's no provision for polygons containing holes.

Общий синтаксис формата такой:

```
[ST] [C] [N] [4] [n] OFF # Заголовок
[Ndim] # Размерность пространства
NVertices NFaces NEdges
x[0] y[0] z[0] # Вершины
...
x[NVertices-1] y[NVertices-1] z[NVertices-1]
Nv[0] v[0][0] v[0][1] ... v[0][Nv-1] color[0] # Грани
...
Nv[NFaces-1] v[NFaces-1][0] ... v[NFaces-1][Nv-1] color[NFaces-1]
```

А теперь по частям. Заголовок с модификаторами:

```
[ST] [C] [N] [4] [n] OFF
```

Каждый указанный модификатор к заголовку указывает, какие данные задаются в точках:

- ST - текстурные координаты

- C - цвет
- N - вектор (нормаль)

Модификатор 4 указывают, что вершины имеют 4 компоненты, 4ая компонента может трактоваться как 4ое измерение или как функция заданная в вершине. Модификатор n указывают, что размерность пространства будет указано отдельно как Ndim. Стоит отметить, что объединение модификаторов 4n задает для каждой точки Ndim+1 компонент. Если указаны модификаторы Nn, размерность векторов в точках равна Ndim, модификатор 4 не влияет на это.

[Ndim]

Опциональный параметр, который используется только с модификатором n, во всех других случаях или отсутствует, или игнорируется.

NVertices NFaces NEdges

Кол-во вершин, граней и рёбер. На данный момент почти все программы игнорируют третий параметр, и обычно его ставят 0.

Далее для каждой вершины идут её координаты и данные связанные с ней.

x y z [nx ny nz] [(c | r g b [a])] [t u]

Где:

- x, y, z — компоненты вершины, при использовании модификаторов 4 или n их может быть иное кол-во.
- nx, ny, nz — компоненты вектора, при модификаторе n их будет Ndim
- c — целочисленный индекс цвета, который следует трактовать как материал
- r, g, b, a — цвет заданный в вершине
- t, u — текстурные координаты

Важно отметить, что порядок данных фиксирован. Это позволяет задавать цвет разными способами. Например задан заголовок STC4OFF, тогда если в строчке 7 чисел, то первые 4 это компоненты точки, последние 2 — текстурные координаты, а оставшиеся одно число - индекс цвета.

Далее для каждой ячейки записываются её размер, список вершин и цвет

`Nv v[0] v[1] ... v[Nv] [(c | r g b [a])]`

Так как в формате OFF ячейки являются многогранниками, тип ячейки напрямую связан с кол-во вершин ячейки: для треугольника $Nv == 3$, для четырехугольника $Nv == 4$.

Trick. Теоритически можно задавать $Nv == 1$ для ячеек типа точек, и $Nv == 2$ для отрезков, но такая возможно официально не документированна, и многие программы будут пропускать такие ячейки.

Если цвет у ячейки не указан, то ячейка помечается цветом по умолчанию из палитры цветов. Многие программы, например Geomview, целочисленный цвет будут преобразовывать в нормальный используя палитру цветов, для Geomview она задается в файле 'smar.fmap' в директории '

(???) Exception: each face's vertex indices are followed by an integer indicating how many color components accompany it. Face color components must be floats, not integer values. Thus a colorless triangular face might be represented as 0. Я пока не нашел таких примеров.

Есть упоминания о наличии бинарного формата, у которого заголово дополняется словом BINARY. Но примеров и описания найти пока не получилось.

Bug. В документации не получилось найти информацию о том какое поведение должно быть если местами цвет указан индексом (материал), а местами цветом с 3 или 4 компонентами.

3 Форматы MESH, BB

Формат Mesh (*.mesh) и присоединенный к нему формат BB (*.bb) - ASCII форматы данных для хранения произвольных сеток и данных на них. Существует бинарная версия.

В этом формате есть возможность записать точки с произвольным кол-вом компонент, ячейки типов вершины, ребра, треугольники, четырехугольники, тетраэдры, гексаэдры. А так же отдельным списком углы (?), рёбра, ...

Нумерация точек и ячеек начинается с 1.

3.1 Из документации по Medit

Структура данных сетки может быть описана очень простым (однако полным) форматом данных. По умолчанию программа (Medit) делает попытку найти mesh файл. Если такого файла не найдено, то производится попытка с другим форматом последовательно сначала с MSH2, а затем с GIS.

Формат MESH

Этот формат состоит из одного (текстового или бинарного) файла. Его структура состоит из нескольких блоков идентифицируемых ключевыми словами. Пробельные символы (space, \r, \n, \t) считаются разделителями элементов. Строка комментариев начинается с символа # и заканчивается концом строки. Комментарии располагаются исключительно между блоками (! могут быть до и после).

Файл должен начинаться со строк

`MeshVersionFormatted 1`

`Dimension 3`

Смысл первого параметра (MeshVersionFormatted) не описывается.

Dimension определяет размерность пространства, то есть задает кол-во компонент для описания точки.

Остальные блоки делятся на обязательные и необязательные. Обязательные поля отвечают за геометрию (координаты/точки) и топологию (элементы сетки), а не обязательные за данные и особенности сетки. Каждый блок записывается по следующему шаблону: ключевое слово, число элементов в блоке, последовательный список компонент каждого элемента. Для простоты чтения этот список разбивают на строки по элементам.

`KeyWord`

```
N
v_1^1 v_1^2 ... v_1^K
...
v_N^1 v_N^2 ... v_N^K
```

Notice. В документации явно указано что используется одинарная точность описания координат и данных.

3.1.1 Точки

Блок с точками начинается с ключевого слова Vertices, потом идёт общее кол-во точек и список точек с их компонентами и ссылкой.

```
Vertices
4
1 2 3 51
0.1 1.2 3.2 72
2.1 3.2 4.5 81
1.e1 22 53 12
9 9 9 11
```

3.1.2 Ячейки

Блоки с ячейками разбиты по типам элементов и начинается с ключевых слов Edges, Triangles, Quadrilaterals, Tetrahedra и Hexaedra, соответственно, потом идёт кол-во ячеек этого типа и их список: индекса вершин ячейки и ссылка. Порядок блоков может быть любой, но блоки одного типа не могут повторяться. Блоки могут отсутствовать.

```
Edges
2
1 2 23
2 3 51
Triangles
2
1 2 3 23
2 3 4 51
Quadrilaterals
2
1 2 3 7 23
2 3 4 8 51
Tetrahedra
```



```

2
1 2 3 4 23
5 3 4 8 51
Hexaедra
2
1 2 3 4 5 6 7 23
4 5 6 7 8 9 10 51
    
```

3.1.3 Особенности

Далее идут блоки ограничивающих (?) элементов или особенностей. В частности, краевые точки (ключевое слово Corner), которые являются C^0 непрерывными (?). Аналогично рёбра (Ridge) как рёбра в которых C^0 непрерывность между смежными гранями. Блоки RequiredVertices, RequiredEdges, RequiredTriangles, RequiredQuadrilaterals, RequiredTetrahedra, RequiredHexaедra задают списки точек и ячеек которые должны сохраняться в при действии алгоритма преобразования сетки. Другими словами эти точки и ячейки выделяются в отдельный список. Структура: ключевое слово, число элементов и список индексов.

```

Corner
3
1 2 3
Ridge
3
41 23 11
RequiredQuadrilaterals
3
42 33 13
    
```

3.1.4 Нормали

Далее идут блоки с описанием нормалей в точках и ячейках и касательных к рёбрам. Сначала идут списки векторов нормалей и касательных с ключевыми словами Normals и Tangents.

```

Normals
3
1.0 0.0 0.0
0.0 1.0 0.0
0.0 0.0 1.0
    
```

Tangents

2

0.0 0.9 0.1

0.5 0.5 0.5

Блок NormalAtVertices задаёт нормали в точках. Элементы этого списка являются пары номер точки - номер нормали. Блок NormalAtTriangleVertices определяет нормали в вершинах треугольников. Элементы этого списка - тройки: номер треугольника, номер точки, номер нормали. Аналогично определяется блок NormalAtQuadrilateralVertices, который задаёт нормали в вершинах квадратов. И TangentAtEdges, который задаёт касательные в рёбрах.

3.1.5 Конец файла

После всех блоков, в конце структура данных должна закрываться ключевым словом End.

3.1.6 Формат bb

4 Формат MSH2

This file format is a very crude data format provided for backward compatibility purposes (the user is strongly encouraged to use the mesh format). Notice that only surface meshes can be described with this format. It is composed of two ASCII files, xxx.points and xxx.faces. The file .points has the following structure :

np

\$x_i y_i z_i ref_i\$

representing the vertex coordinates in single precision and the vertex reference. The file .faces represents the mesh topology and contains the following records :

nf

\$d v_i^1 v_i^2 v_i^3 ref_i ref_i^1 ref_i^2 ref_i^3 ref_i^4\$

where d is either 3 (triangles) or 4 (quad), v_i^j represents a vertex number, ref_i sub-domain containing the face and ref_i^j are the edge references.

4.1 Формат GIS

This file format is used to import terrains (or cartesian surfaces). It consists of a single file (ASCII or binary). The file structure is composed of a header (ASCII) indicating the terrain resolution and spatial location :

```
G_i  
sx sy  
cx cy cz  
gu hu  
xmi ymi
```

```
G_i  
sx sy  
cx cy cz  
gu hu  
xmi ymi
```

where G_i indicates the file type (G_1 for an ascii file and G_2 for a binary file), $sx\ sy$ are two integers defining the terrain resolution (i.e., the size of the underlying grid), $cx\ cy\ cz$ represent 3 scaling factors, $gu\ hu$ are the model units and $xmi\ ymi$ represent the location of the lower left corner of the terrain. Then, follows a series of scalar (single float) values figuring the heights of each node of the grid.

5 Формат OOGL

6 Формат VTK

7 Формат NEU

8 Формат DAT/TEC

9 Формат IGES

10 Список форматов

- OFF
- MESH
- OBJ
- VTK
- NEU
- DAT/TEC
- IGES
- Прочие:
 - QUAD
 - BBOX
 - BBP
 - BEZ
 - VECT
 - SKEL
 - LIST
 - TLIST
 - GROUP
 - DISCGRP
 - COMMENT

- PLY
- BYU
- X3D
- JVX
- VRML
- Maya
- POV
- LWO
- 3DS
- RIB
- DXF
- STL
- ZPR
- HDF
- HDF5
- NASACDF
- NetCDF
- FITS
- SP3
- DICOM
- Affymetrix
- BDF
- EDF
- MOL
- SDF

- SMILES
- PDB
- GenBank
- FASTA
- NDK
- GRIB

11 Список программ

- OnReader
- Paraview
- Ansys Fluent
- Ansys Icem
- 3D max
- Grapher
- Tecplot
- msh
- CGNS