УДК 519.6

# ФОРМАТ ДЛЯ ОПИСАНИЯ НЕРЕГУЛЯРНОЙ МНОГОГРАННОЙ СЕТКИ ПРОИЗВОЛЬНОЙ СТРУКТУРЫ В МЕТОДИКЕ ТИМ

А. А. Воропинов, С. С. Соколов, А. И. Панов, И. Г. Новиков
(РФЯЦ-ВНИИЭФ)

Рассматриваются три формата представления нерегулярной многогранной сетки произвольной структуры применительно к реализации в рамках методики ТИМ. Приводятся описания структур данных, их преимущества и недостатки, оценки требуемой памяти на примере сеток трех типов (шестигранной, на основе тел Вороного и на основе шестиугольных призм).

## Введение

Один из важных вопросов, возникающих при разработке любой методики, базирующейся на использовании конечных разностей, — тип расчетной сетки. Применительно к методикам, использующим нерегулярные сетки, имеются различные подходы.

Один из самых распространенных — использование нерегулярных сеток, состоящих из ячеек одинаковой формы. Этот подход, как правило, применяется для тетраэдральных и шестигранных сеток (все грани которых четырехугольные). Ряд структур данных для описания таких сеток приведен в работе [1].

Другой широко известный подход — использование в сетке ограниченного набора тел, например шестигранников, с различными возможностями вырожденности (отсутствия у ячейки одной или нескольких граней или ребер) [2]. Одна из разновидностей такого подхода строится на описании тела в зависимости от его формы, т. е. для каждого вида тела имеется своя министруктура (по такому принципу реализованы, например, сеточная библиотека Mesh Toolkit (MSTK) [3] и библиотека визуализации VTK [4]). На сетку можно наложить и такое ограничение, как фиксированное количество *соседей* в узле. Сетка с подобной структурой используется, например, в методике ТМК [5], где внутренний узел всегда окружают 4 ячейки и в узле сходятся 4 ребра. Помимо указанных, часто на сетку накладываются дополнительные ограничения, например, использование трехгранных углов у ячеек.

Основное требование к формату описания структуры сетки — его полнота, т. е. возможность получить информацию о *соседстве* для любого элемента сетки. При этом одновременно нужно решать две во многом противоположные задачи:

- данные о структуре сетки для одной счетной точки должны занимать как можно меньше оперативной памяти ЭВМ;
- информацию о соседстве для элемента сетки нужно получать с высокой скоростью.

В реальных условиях эти две задачи одновременно решить сложно, поэтому формат хранения необходимо выбирать, исходя из того, к каким данным приходится обращаться наиболее часто при проведении расчетов.

В методике ТИМ [6] используются нерегулярные многогранные сетки произвольной структуры: ячейки состоят из произвольного количества граней, грани содержат произвольное количество узлов, в узлах может сходиться произвольное количество ребер. В ряде работ предлагаются разные способы хранения данных для описания сеток произвольной структуры [1, 7, 8]. Однако во всех случаях либо имеется большая избыточность хранимой информации, что приводит к резкому увеличению требуемой памяти, либо предлагаемые структуры данных являются неоптимальными для реализации алгоритмов методики ТИМ.

В данной статье в первую очередь речь идет о структуре рабочих данных, хранящихся в оперативной памяти ЭВМ, и использовании этой структуры при реализации счетных алгоритмов для получения информации о соседстве элементов сетки.

Необходимо отметить, что выбор формата хранения данных в значительной степени влияет на сложность алгоритмов и скорость выполнения программ. Структура данных разрабатывается в самом начале программной реализации методики. Дальнейшие ее изменения могут потребовать существенных доработок программ или даже пересмотра их алгоритмов. По этой причине выбор подходящей и удобной для использования структуры является очень важным аспектом.

## 1. Методика ТИМ

Прежде чем сформулировать требования, предъявляемые к структуре данных при реализации методики ТИМ, приведем краткое описание методики.

Методика ТИМ предназначена для расчета трехмерных задач механики сплошных сред на многогранных неструктурированных лагранжевых сетках. Она позволяет проводить расчеты на сетках с произвольным количеством связей в узлах (примыкающих к узлу ячеек, ребер и граней). Методика применима как на нерегулярных многогранных сетках на базе тел Вороного (в каждом узле сетки сходятся четыре ребра), так и на шестигранных сетках (в каждом узле сходятся шесть ребер) и нерегулярных многогранных сетках, имеющих произвольное количество *соседств* в узле (в частности пять). Для всех типов сеток используется единый счетный алгоритм. При решении уравнений механики сплошных сред используется декартова система координат.

В методике ТИМ применяется расщепление по процессам и к настоящему моменту реализованы приближения для решения следующих классов задач: газовой динамики, упругопластичности, теплопроводности, магнитной гидродинамики, двухтемпературности, двухпотоковости. При расчете задачи используются программы автоматической локальной перестройки сетки путем дробления или объединения ячеек, что приводит к изменению структуры сетки в процессе расчета.

Алгоритмы расчета процессов газовой динамики и упругопластичности, как правило, построены *по ячейкам* (имеется в виду способ организации счетных циклов). Алгоритмы расчета диффузионных процессов чаще используют *узловую* организацию.

Методика ТИМ предполагает трехуровневое распараллеливание: на первом (верхнем) уровне — распараллеливание по счетным областям, на втором — мелкозернистое распараллеливание по параобластям внутри счетной области. На этих двух уровнях используется модель распределенной памяти. На третьем (нижнем) уровне распараллеливание производится внутри параобласти в модели общей памяти. При мелкозернистом распараллеливании декомпозиция осуществляется по ячейкам с наложением в один слой между параобластями.

При разработке основ методики перед авторами возникла проблема выбора формата хранения данных таким образом, чтобы он подходил для реализации всех алгоритмов и позволял описывать сетки произвольной структуры. При этом структура данных должна быть достаточно экономичной, т. е. требовать минимального объема оперативной памяти, и должна быстро предоставлять информацию о соседстве элементов сетки для счетных алгоритмов, выполняемых как в последовательном, так и в параллельном режиме.

## 2. Структуры данных

При выборе формата хранения информации для сетки методики ТИМ были рассмотрены три структуры данных. Основываясь на основных элементах этих структур, их можно определить следующим образом:

1. Структура хранения *по граням*: для ячейки хранится список ее граней, для грани — список номеров узлов.
2. Структура хранения *по углам*: для узла хранится список окружающих ячеек и согласованный с ним список многогранных углов.
3. Структура *ребро-грани*: в качестве основного элемента выступает грань вместе с одним из ее ребер.

По предварительным оценкам, при решении уравнений газодинамики и упругопластичности достаточно "быстрой" и удобной для получения необходимой информации о сетке является структура хранения по граням. Для решения уравнения теплопроводности более быстрым является формат ребро-граней. Обе эти структуры позволяют достаточно быстро получать информацию о связях, необходимую при построении функционалов взаимодействия для поддержания приемлемой счетной сетки методом наложения дифференцируемых связей [9].

Рассмотрим более подробно перечисленные структуры данных, их преимущества и недостатки.

## 3. Структура хранения по граням

Алгоритмы, используемые при решении задач газодинамики и упругопластичности, построены таким образом, что расчет уравнений движения и энергии может выполняться по ячейкам. При этом рассчитывается взаимодействие между ячейками через грань, а информация о соседстве в узлах для проведения расчета является неосновной.

С учетом вышесказанного формат хранения сетки можно представить следующим образом:

1. Для ячейки хранится неупорядоченный список номеров всех ее граней.
2. Для грани хранятся:
   1) номера двух ячеек, ее формирующих; для грани, лежащей вдоль границы, первым номером является номер граничного условия;
   2) список узлов, формирующих грань, причем со стороны первой формирующей грань ячейки узлы упорядочены против часовой стрелки, а со стороны второй ячейки — по часовой стрелке.
3. Для узла хранится номер одной из граней, сходящихся в данном узле, для граничного узла — номер одной из граничных граней.

Таким образом, связь между элементами сетки можно схематично представить следующим образом:

$$\text{ячейка} \longleftrightarrow \text{грани} \longrightarrow \text{узлы} \longrightarrow \text{грань}.$$

Положительным свойством рассматриваемого формата хранения является то, что элементы структуры соответствуют реальным элементам сетки. Кроме того, информация о ячейках и гранях полностью содержится в структуре и нет необходимости выполнять дополнительные действия для ее получения. Это, в свою очередь, позволяет сократить количество дополнительных операций при работе с этой структурой и повысить скорость расчетов (например, при решении уравнений газодинамики и упругопластичности). Однако для алгоритмов, построенных по узлам, необходима информация о соседстве в узле. С этой целью может либо выполняться операция локального поиска, либо (для ускорения) формироваться избыточная дополнительная информация, рассмотренная ниже. Такие алгоритмы в методике ТИМ используются для диффузионных процессов, например, при решении уравнений теплопроводности и магнитной гидродинамики.

При распараллеливании структура хранения по граням хорошо подходит для организации декомпозиции по ячейкам.

**Обобщение структуры хранения по граням на двумерный случай.** Одним из важных преимуществ структуры хранения по граням является возможность ее обобщения на двумерный случай. В этом случае грань превращается в ребро.

Формат хранения по граням для двумерного случая можно представить следующим образом:

1. Для ячейки хранится список ребер в порядке обхода против часовой стрелки.
2. Для ребра хранятся:
   1) номера двух ячеек, его формирующих; для ребра, лежащего вдоль границы, первым номером является номер граничного условия;
   2) номера двух узлов, которые соединяет ребро, причем со стороны первой формирующей ребро ячейки узлы упорядочены против часовой стрелки, а со стороны второй ячейки — по часовой.
3. Для узла хранится номер одного из ребер, сходящихся в данном узле, для граничного узла — номер правого граничного ребра, если смотреть со стороны границы.

Данная структура является не самой экономичной по требуемой оперативной памяти, однако ее использование позволяет проводить разработку некоторых алгоритмов и программ сразу для двумерного и трехмерного случаев. Такая структура хранения используется в методике ТИМ-2D [10].

**Пополнение структуры хранения по граням информацией о многогранных и двугранных углах.** Для хранения многогранных углов достаточно ввести еще один элемент — угол, для которого хранить (рис. 1) номер ячейки $N$, номер узла — вершины $I$ и номера узлов

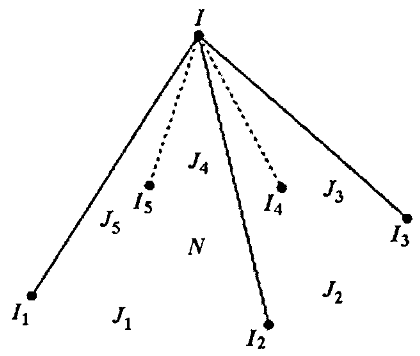


Рис. 1. Информация об угле

основания (противоположных концов ребер этого угла) в порядке против часовой стрелки: $I_1$, $I_2, \ldots, I_k$ (в данном случае $k = 5$). Для формирования двугранных углов, кроме узлов основания (или вместо них), надо хранить номера граней $J_1, J_2, \ldots, J_k$, сходящихся в этом узле.

**Пополнение структуры информацией о ребро-гранях.** Для введения структуры *ребро-грани* рассмотрим более детально схему хранения информации в памяти. Так, для каждой грани хранится список номеров узлов. Понятно, что у разных граней количество узлов различно, поэтому для их хранения необходимо вводить списки переменной длины. Для реализации списковой структуры используются один большой одномерный массив, непосредственно содержащий списки узлов для всех граней области (контейнерный массив — разд. 6).

Предположим, что общая размерность контейнерного массива равна $EF$. Рассмотрим грань с номером $k$. Пусть эта грань содержит $n$ узлов и, начиная с элемента с номером $i$, до элемента $i + n - 1$ в контейнерном массиве располагается информация об этой грани. Рассмотрим элементы контейнерного массива с индексами $j$ и $j + 1$ ($i \le j < i + n - 1$). В этих двух элементах записаны номера двух узлов грани $k$, причем они упорядочены в последовательности обхода против часовой стрелки для первой ячейки этой грани. То есть данные узлы формируют ребро с номером $j - i + 1$ грани $k$. Но поскольку информация о грани в массиве располагается только в одном месте, то соответствующая пара чисел и дает *ребро-грань*. Наиболее естественно присвоить этой ребро-грани номер $j$. Однако таким образом для грани $k$ получается $n - 1$ ребро-грань, в то время как реально она содержит $n$ ребро-граней. Последняя ребро-грань определяется парой элементов массива с индексами $i + n - 1$ и $i$. Таким образом, получается полная нумерация ребро-граней. Кроме того, такая нумерация позволяет автоматически определить две соседние ребро-грани, лежащие в той же грани. Для ребро-грани с номером $j$ таковыми являются ребро-грани с номерами

$$\begin{array}{llll} j-1 & \text{и} \;\; j+1, & \text{если} & i<j<i+n-1; \\ i-n+1 & \text{и} \;\; j+1, & \text{если} & j=i; \\ j-1 & \text{и} \;\; i, & \text{если} & j=i-n+1. \end{array}$$

Однако в структуре ребро-грани нужна еще информация о двух соседних ребро-гранях (разд. 4). Для этих элементов необходимо размещение дополнительных массивов размерностью $EF$. Кроме того, нужен массив с такой же размерностью, где для каждой ребро-грани хранится номер грани, к которой она относится, а также массив, в котором для каждого узла хранится номер опорной ребро-грани. Эти дополнительные массивы требуются для выполнения алгоритмов, использующих структуру ребро-грани без непосредственного участия структуры хранения по граням (например, для построения циклов по ребро-граням или для перехода от узла к ребру).

## 4. Описание произвольной нерегулярной трехмерной сетки с помощью ребро-граней

В дальнейшем будем обозначать буквой $I$ номера узлов, буквой $N$ — номера ячеек, $J$ — номера граней, $K$ — номера ребро-граней счетной сетки. Ребро-грань — это ребро счетной сетки вместе с прилегающей к нему гранью. Понятно, что с каждым ребром сетки связано столько ребро-граней, сколько граней опирается на данное ребро. А каждой грани соответствует столько ребро-граней, сколько ребер имеет данная грань.

На рис. 2 изображена ориентированная грань $J$, разделяющая ячейки $N_1$ и $N_2$. Обход грани $J$ осуществляется против часовой стрелки, если смотреть на грань из ячейки с номером $N_1$. Через $a$ обозначено ребро грани $J$, выходящее из узла $I$ по направлению обхода грани. Ребро-грань $K$ соответствует ребру $a$ и грани $J$. Ребро-грани $K_1$ и $K_2$ связаны с ребром $a$ и с соседними к $J$ гранями, расположенными в ячейках $N_1$ и $N_2$ соответственно. Через $c$ и $b$ обозначены предыдущее и последующее по

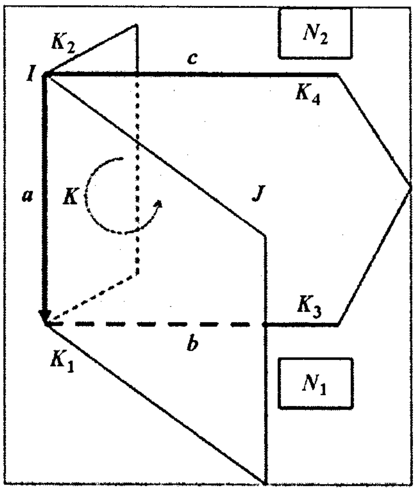


Рис. 2. Соседи ребро-грани $K$

отношению к $a$ ребра, принадлежащие грани $J$. Ребро-грани $K_3$ и $K_4$ связаны с гранью $J$ и ребрами $b$ и $c$ соответственно.

Пусть $F$ — число граней, а $EF$ — число ребро-граней счетной сетки. Тогда нерегулярную трехмерную сетку произвольной структуры можно описать массивами, приведенными в табл. 1. Помимо этих массивов, для каждой грани, ячейки и узла хранится номер одной из примыкающих ребро-граней.

Рассматриваемый формат хранения обладает свойством *фиксированности* — для всех элементов структуры хранится одинаковое количество величин вне зависимости от сетки. Это позволяет не использовать списки переменной длины и несколько облегчает дробление и объединение ячеек. Хотя для получения информации о соседстве основных элементов сетки необходимо выполнение дополнительных операций, некоторые из них не требуют локального поиска и выполняются с высокой скоростью.

Структура ребро-грани может быть расширена информацией о гранях и ячейках. Для этого нужно хранить структуру по граням, кроме опорных граней для узлов, т. е. для ячеек хранить список граней, для граней — список узлов. Отметим, что при таком расширении структура теряет свойство фиксированности и для нее нужна организация контейнерных массивов (разд. 6).

## 5. Структура хранения по углам

В структуре хранения по углам сетка описывается через соседство в узлах. В качестве элементов этой структуры выступают списки номеров:

1) ячеек $N_1$, $N_2$, ..., $N_n$, окружающих узел;
2) узлов основания многогранного угла с вершиной в рассматриваемом узле, расположенных в определенном порядке обхода для каждой ячейки, окружающей этот угол.

В результате для каждого узла $i$ хранятся:

$k_0$ — количество ячеек (в том числе граничных условий) в узле $i$;

$n_1$ — номер первой такой ячейки;

$k_1$ — количество ребер, образующих многогранный угол ячейки $n_1$ в узле $i$;

$n_{1,1}$, $n_{1,2}$, ..., $n_{1,k_1}$ — номера узлов на противоположных концах этих ребер;

$n_2$ — номер второй ячейки в узле $i$;

$k_2$ — количество ребер, образующих многогранный угол ячейки $n_2$ в узле $i$;

$n_{2,1}$, $n_{2,2}$, ..., $n_{2,k_2}$ — номера узлов на противоположных концах этих ребер;

.......................................

$n_{k_0}$ — номер $k_0$-й ячейки в узле $i$;

$k_{k_0}$ — количество ребер, образующих многогранный угол ячейки $n_{k_0}$ в узле $i$;

$n_{k_0,1}$, $n_{k_0,2}$, ..., $n_{k_0,k_{k_0}}$ — номера узлов на противоположных концах этих ребер.

Узлы на противоположных концах ребер многогранного угла нумеруются в порядке положительного обхода (против часовой стрелки) (рис. 3). В данном случае также необходима

*Таблица 1*

**Массивы, описывающие сетку при помощи ребро-граней**

| Номер грани | Номера соседних ячеек $N_1, N_2$ | | Номер ребро-грани | Номера соседей для ребро-грани: узла | грани | ребро-граней $K_1, \ldots, K_4$ | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | $N_{1,1}$ | $N_{2,1}$ | 1 | $I_1$ | $J_1$ | $K_{1,1}$ | $K_{2,1}$ | $K_{3,1}$ | $K_{4,1}$ |
| 2 | $N_{1,2}$ | $N_{2,2}$ | 2 | $I_2$ | $J_2$ | $K_{1,2}$ | $K_{2,2}$ | $K_{3,2}$ | $K_{4,2}$ |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| $F$ | $N_{1,F}$ | $N_{2,F}$ | $EF$ | $I_{EF}$ | $J_{EF}$ | $K_{1,EF}$ | $K_{2,EF}$ | $K_{3,EF}$ | $K_{4,EF}$ |

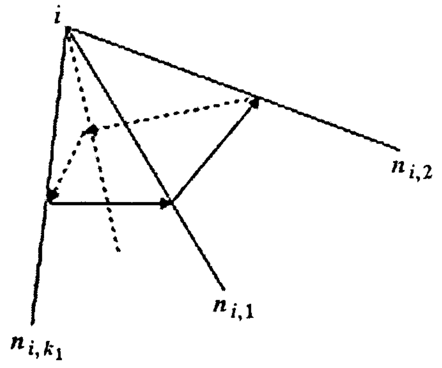


Рис. 3. Многогранный угол

реализация списков переменной длины (т. е. контейнерных массивов) для хранения последовательностей этих узлов (разд. 6).

Для каждой ячейки хранится номер одного из ее узлов (опорный узел).

Структура хранения по углам хорошо подходит для реализации алгоритмов, использующих соседство в узле и декомпозицию по узлам при мелкозернистом распараллеливании.

Недостатком структуры является невозможность описать некоторые особенности сетки, например узлы, в которых сходится по два ребра. Для алгоритмов по ячейкам или по граням такая структура оказывается малоэффективной, ее приходится расширять дополнительной информацией, например, о гранях и ячейках. При этом нужно хранить структуру по граням, кроме опорных граней для узлов.

## 6. О хранении списковых массивов переменной длины

Как видно из описания, структуры хранения данных по граням и по углам требуют организации списковых массивов переменной длины, т. е. массивов, количество элементов в которых различно для разных элементов сетки. Для структуры хранения по граням это два списка: грани для ячеек и узлы граней. Для структуры хранения по углам это список узлов на противоположных концах ребер, образующих многогранный угол. Решать эту проблему можно двумя способами.

Первый способ заключается в фиксировании определенного количества элементов сетки. В случае, если количество заполненных элементов меньше, чем отведено, можно использовать дополнительный признак, например хранение нуля вместо отсутствующих элементов. Однако такой подход, во-первых, вводит ограничение на сетку, во-вторых, влечет за собой неоптимальное использование оперативной памяти.

Второй подход, который и был выбран для представления указанных списков, заключается в хранении списковых массивов по следующей схеме:

– организуется большой контейнерный массив (контейнер), который непосредственно содержит списковые массивы переменной длины;
– дополнительно формируются два вспомогательных массива, в которых хранятся адрес смещения в контейнере и количество элементов списка.

В принципе можно использовать один дополнительный массив смещений (количество элементов получается как разность смещений следующего и текущего списков). Однако в результате выполнения программ локальных перестроек структура сетки, а значит количество элементов в списках, меняется, следовательно, необходимо выполнять сдвижки и раздвижки контейнерного массива, что при частом использовании значительно увеличивает время счета. Организация отдельного массива размеров списков и обновление его по мере надобности решают данную проблему. Если количество элементов списка увеличивается, этот список переносится в конец контейнера, а старое место становится *дырой*. При увеличении количества дыр контейнер можно уплотнить. На практике процедура уплотнения выполняется один раз за несколько десятков или даже сотен шагов и практически не влияет на скорость выполнения программы.

## 7. Оценка затрат памяти для различных структур данных

Чтобы окончательно выбрать формат для описания сетки, были выполнены оценки затрат памяти для всех рассмотренных структур данных на примере сеток на базе тел Вороного [11], сетки на основе шестиугольных призм и шестигранной сетки (регулярной). Эти сетки являются базовыми для расчетов по методике ТИМ.

Требование к памяти — одно из основных при выборе формата для описания трехмерной сетки. Неэкономичная структура данных может занимать память, на порядок большую, чем необходима для хранения рассчитываемых в задаче величин.

Оценки производились в предположении, что сетка заполняет все пространство, т. е. не учитывались граничные элементы, которые в реальных сетках всегда присутствуют. Приведенные замеры хорошо соответствуют реальным сеткам, состоящим из нескольких тысяч ячеек и более (для сеток, состоящих из 1 млн. ячеек погрешность оценок составляет менее 5 %). При подсчете памяти для структур хранения по углам и по граням учитывалась дополнительная память, необходимая для организации контейнеров.

Введем следующие обозначения, где прописными буквами обозначим суммарные количества элементов в сетке, малыми — количества величин для описания одного элемента (эти количества являются усредненными и поэтому могут быть дробными числами):

$N$ — общее количество узлов сетки;

$b = 1$ — наличие одного опорного элемента, например опорного узла для ячейки;

$e = 2$ — количество ячеек, формирующих грань;

$f = 2$ — наличие двух дополнительных массивов для описания структуры ребро-грани через расширение структуры хранения по граням;

$g = 3$ — количество узлов в основании многогранного угла (в рассматриваемых сетках всегда равное 3, но в общем случае произвольное);

$h = 6$ — количество элементов, необходимых для описания ребро-грани;

$i = 2$ — наличие дополнительной памяти, необходимой для организации контейнерных массивов (количество элементов и смещение от начала массива);

$n$ — количество ячеек, окружающих узел, или количество углов с общим узлом-вершиной;

$k$ — количество граней одной ячейки;

$l$ — количество ребер, сходящихся в узле;

$m$ — среднее количество узлов одной грани (для одной ячейки);

$j$ — количество узлов одной ячейки.

Тогда суммарные величины по сетке выражаются следующими формулами:

$K \approx \dfrac{Nn}{j}$ — общее количество ячеек;

$F \approx K\dfrac{k}{e} = N\dfrac{nk}{2j}$ — общее количество граней;

$E \approx Fm = N\dfrac{nkm}{2j}$ — общее количество ребро-граней;

$L \approx Nn$ — общее количество многогранных углов.

Характерные параметры для разных сеток приведены в табл. 2. В таблице указаны размеры памяти, необходимой для хранения информации о связях для внутренних узлов (без учета граничных ячеек). За единицу измерения принято количество узлов. Для лучшего понимания на рис. 4 приведены примеры ячеек разных типов сеток и соответствующих связей узлов.

*Таблица 2*

**Параметры сеток разных типов**

| Тип ячеек сетки | Параметры сетки, $N$ | | | | Параметры элемента сетки | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $K$ | $F$ | $E$ | $L$ | $n$ | $k$ | $l$ | $m$ | $j$ |
| Тела Вороного | $\frac{1}{6}$ | $\frac{7}{6}$ | 6 | 4 | 4 | 14 | 4 | $5\frac{1}{7}$ | 24 |
| Шестиугольные призмы | $\frac{5}{12}$ | $1\frac{2}{3}$ | $7\frac{1}{2}$ | 5 | 6 | 8 | 5 | $4\frac{1}{2}$ | 12 |
| Шестигранники | 1 | 3 | 12 | 8 | 8 | 6 | 6 | 4 | 8 |

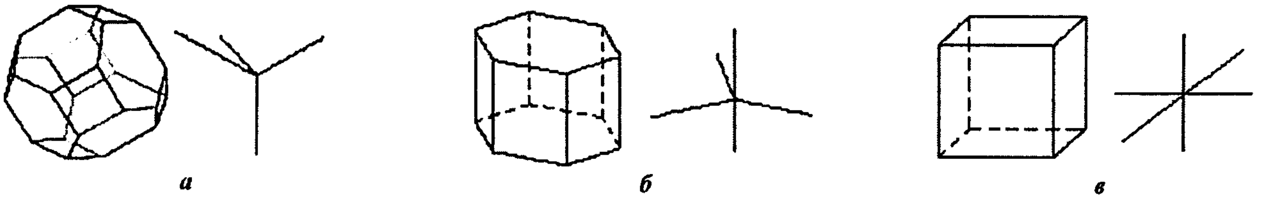

Рис. 4. Примеры ячеек и связей узлов: *а* — сетка на базе тел Вороного; *б* — сетка на основе шестиугольных призм; *в* — шестигранная сетка

Приведем формулы расчета необходимой памяти для различных структур данных ($M$ — общая размерность всех массивов для рассматриваемой структуры):

1. Для структуры хранения по граням
$$M^F = (k+i)\,K + bN + (m+e+i)\,F.$$
Память, необходимая для дополнительного формирования ребро-граней,
$$M_E^F = (f+b)\,E + bN.$$
Память, необходимая для дополнительного хранения углов, представленных в виде узлов или граней,
$$M_{L1}^F = (n+i)\,N + (i+b+g)\,L.$$
Память, необходимая для дополнительного хранения углов, представленных в виде узлов и граней одновременно,
$$M_{L2}^F = M_{L1}^F + gL.$$
2. Для структуры хранения по углам
$$M^L = 2bK + (b+e)F + (n+i)N + (i+b+g)L.$$
Память, необходимая для дополнительной информации о ячейках и гранях,
$$M_F^L = (k+i)K + (m+e+i)F.$$
3. Для структуры ребро-грани
$$M^E = hE + b\,(N+K+F) + eF.$$
Память, необходимая для дополнительной информации о ячейках и гранях,
$$M_F^E = (k+i)K + (m+i)F.$$

Результирующие параметры размеров памяти приведены в табл. 3. За единицу измерения взято количество узлов.

Как видно из табл. 3, наиболее экономичной является структура хранения по граням без использования дополнительной информации. Главным образом, поэтому для методики ТИМ было решено выбрать структуру хранения по граням.

## Заключение

Рассмотрены три существенно различные структуры данных для описания нерегулярных многогранных сеток. В каждой из структур использованы разные базовые элементы, определившие названия этих структур, — *по граням*, *ребро-грани* и *по углам*. На основе сделанных оценок экономичности требуемой памяти и простоты реализации алгоритмов для методики ТИМ выбрана структура хранения по граням.

Помимо отмеченных выше характеристик, разработанная структура хранения по граням имеет следующие положительные аспекты:

1. Наиболее естественное описание структуры сетки на основе геометрических элементов (ячеек, граней, узлов), что облегчает ее восприятие и, как следствие, упрощает написание программ.
2. Естественное обобщение структуры на двумерный случай, что позволяет разрабатывать некоторые алгоритмы и программы сразу для трехмерного и двумерного случаев.

Структура хранения по граням имеет один недостаток: в ней доступ к сведениям о соседстве узлов и получение двугранных и многогранных углов являются наиболее дорогостоящими. Од-

*Таблица 3*

**Размеры необходимой памяти для описания сеток разных типов с использованием различных структур данных**

| Тип ячеек сетки | Структура | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | по граням | | | | по углам | | ребро-грани | |
| | $M^F$ | $M^F+M^F_{L1}$ | $M^F+M^F_{L2}$ | $M^F+M^F_E$ | $M^L$ | $M^L+M^L_F$ | $M^E$ | $M^E+M^E_F$ |
| Тела Вороного | $13\frac{1}{3}$ | $43\frac{1}{3}$ | $55\frac{1}{3}$ | $32\frac{1}{3}$ | $33\frac{5}{6}$ | $46\frac{1}{6}$ | $40\frac{2}{3}$ | $50\frac{2}{3}$ |
| Шестиугольные призмы | 23 | 67 | 85 | 51 | 51 | 73 | $61\frac{1}{2}$ | $79\frac{1}{2}$ |
| Шестигранники | 33 | 91 | 115 | 70 | 69 | 101 | 83 | 109 |

нако при выполнении счетных алгоритмов время получения этих данных без использования дополнительной информации оказалось удовлетворительным.

Разработанная структура, помимо реализации в рамках методики ТИМ, была взята за основу для представления нерегулярной сетки в библиотеке ЕФР [12] (из структуры исключен вспомогательный массив опорных граней для узлов).

## Список литературы


1. *Beall M. W., Shephard M. S.* A general topology-based mesh data structure // Int. J. Num. Meth. in Eng. 1997. Vol. 40. P. 1573—1596.
2. *Ушакова О. В.* Классификация шестигранных ячеек // Прикладная геометрия, построение расчетных сеток и высокопроизводительные вычисления. Тр. Всерос. конф. Москва, 4—7 июля 2006 г. / Под ред. Ю. Г. Евтушенко, М. К. Керимова, В. А. Гаранжа. М.: ВЦ им. А. А. Дородницына РАН, 2006. С. 172—179.
3. *Garimella R. V.* MSTK — A flexible infrastructure library for developing mesh based applications // 13th Int. Meshing Roundtable. USA, Williamsburg: VA, 2004.
4. *Schroeder W., Martin K., Lorensen W.* The Visualization Toolkit. An Object Oriented Approach to 3D Graphics. USA: Prentice-Hall, 2004.
5. *Еременко А. Ю., Мотлохов В. Н., Рассказова В. В., Софронов И. Д.* Методика решения задач трехмерной нестационарной газовой динамики на нерегулярных лагранжевых сетках // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 1998. Вып. 4. С. 44—57.
6. *Соколов С. С., Панов А. И., Воропинов А. А., и др.* Методика ТИМ расчета трехмерных задач механики сплошных сред на неструктурированных многогранных лагранжевых сетках // Там же. 2005. Вып. 3. С. 37—52.
7. *Garimella R.* Mesh data structure selection for mesh generation and FEA applications // Int. J. of Num. Meth. in Eng. 2002. Vol. 55, № 4. P. 441—478.
8. *Степанов А. А.* Средства разработки программного обеспечения в приложении к расчетным сеткам // Прикладная геометрия, построение расчетных сеток и высокопроизводительные вычисления. Тр. Всерос. конф. Москва, 4—7 июля 2006 г. / Под ред. Ю. Г. Евтушенко, М. К. Керимова, В. А. Гаранжа. М.: ВЦ им. А. А. Дородницына РАН, 2006. С. 172—179.
9. *Новиков И. Г., Панов А. И., Соколов С. С.* Способ коррекции нерегулярной лагранжевой сетки методом наложения дифференцируемых связей // Журнал вычисл. мат. и мат. физ. 2005. Т. 45, № 8. С. 1487—1500.
10. *Соколов С. С., Воропинов А. А., Новиков И. Г. и др.* Методика ТИМ-2D для расчета задач механики сплошной среды на нерегулярных многоугольных сетках с произвольным количеством связей в узлах // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2006. Вып. 4. С. 29—43.
11. *Вороной Г. Ф.* Собрание сочинений. Киев: Изд-во АН УССР, 1952.
12. *Волгин А. В., Тарасов В. И., Красов А. В., Кузнецов М. Ю.* Единый Файловый Разрез (ЕФР) для универсального представления расчетных данных // Прикладная геометрия, построение расчетных сеток и высокопроизводительные вычисления. Тр. Всерос. конф. Москва, 28 июня—1 июля 2004 г. М.: ВЦ им. А. А. Дородницына РАН, 2004. Т. 2. С. 220—225.


Статья поступила в редакцию 16.07.07.