УДК 519.6

ПРЕПОСТПРОЦЕССОР ЛОГОС-ПРЕПОСТ. ОСОБЕННОСТИ МЕХАНИЗМА ОТБОРА ЭЛЕМЕНТОВ

Д. В. Антошин, А. В. Гордеев
(РФЯЦ-ВНИИЭФ, г. Саров)

Описан механизм отбора элементов расчетной сетки, реализованный в препостпроцессоре ЛОГОС-Препост. Представлены две модели отбора элементов: старая, использующая VTK-фильтры, и новая, основанная на механизме операций. Перечислены недостатки старой модели. Описаны особенности новой модели, позволившие значительно повысить быстродействие и расширить функциональность; проведено ее сравнение с зарубежными аналогами по скорости отбора сеточных элементов.

Ключевые слова: препостпроцессор, пикинг, VTK, механизм операций, визуализация.

Введение

Препостпроцессор (ППП) ЛОГОС-Препост предназначен для построения трехмерных сеток, необходимых программам инженерного анализа ЛОГОС-Прочность, ЛОГОС-Аэрогидромеханика, ЛОГОС-ДАНКО, ЛОГОС-Адаптивность, для подготовки и запуска счета с помощью этих методик, предварительной обработки сеточных моделей (предобработка) и визуализации результатов счета (постобработка). ЛОГОС-Препост представляет собой пакет программных модулей, дающих пользователю удобный интерфейс для выполнения указанных задач. Актуальной на данный момент является версия 4.0.6.

Одним из важных компонентов препостпроцессора ЛОГОС-Препост является механизм отбора элементов расчетной сетки. Под элементами подразумеваются наборы узлов, граней, ячеек или подобластей. Отобранные элементы могут быть использованы в других модулях ППП, например, для визуализации, модификации сетки, задания начальных и граничных условий. Поэтому важно, чтобы система отбора элементов обладала большим количеством доступных операций. Дополнительным требованием, предъявляемым к этой системе, является скорость отбора элементов.

В данной статье представлены две модели отбора элементов: первая, использующая фильтры VTK [1], и вторая, основанная на механизме операций. Описаны недостатки первой модели и особенности второй.

Механизм отбора элементов в ППП

Для реализации механизма отбора элементов сетки разработан специальный комплекс средств, управляемый пользователем посредством диалога *Наборы данных* (рис. 1).

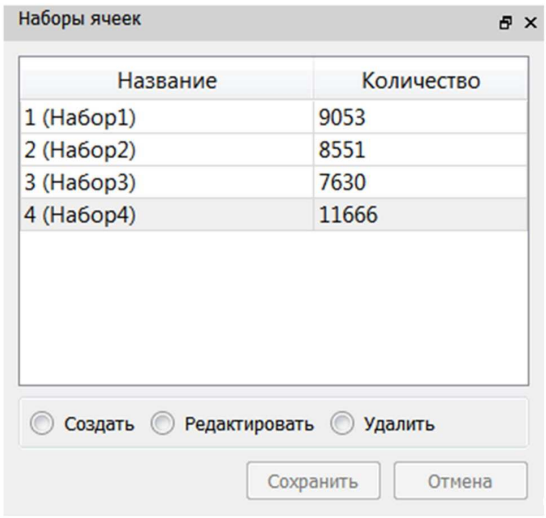

Рис. 1. Внешний вид диалога *Наборы данных* при работе с ячейками сетки

Диалог *Наборы данных* предназначен для создания и управления наборами элементов и может осуществляться в четырех режимах в соответствии с отбираемыми элементами. В зависимости от модели решаемой задачи некоторые режимы могут быть недоступны. Начиная с версии 4.0.6 пользователь может просматривать или удалять несколько наборов одновременно, а также создавать наборы граней по наборам узлов.

При запуске механизма отбора элементов сетки слева от окна визуализации появляется панель управления режимами пикинга (рис. 2). Под пикингом подразумевается способ отбора элементов сеточной модели интерактивным указанием с помощью мыши. С помощью панели пикинга пользователь может изменять параметры отбора элементов. Для изменения действия кнопок, имеющих в правом нижнем углу черный треугольник, необходимо нажать на кнопку и подождать 1—2 секунды для появления выпадающего меню.

В текущей версии доступны следующие возможности по установке режимов пикинга:

- выбор вида области выделения объектов сетки: точка, прямоугольник, круг, многоугольник. На рис. 3 приведен пример использования разных областей выделения элементов сеток;
- выбор режима отбора объектов сетки. Можно установить, где выбирать элементы: внутри, снаружи или на пересечении области выделения;
- добавление/удаление отобранных элементов из текущего набора;
- отбор объектов сетки по углу между ними (рис. 4). При работе с узлами сетки доступен режим отбора узлов по углу между ребрами;
- подсветка объектов сетки до выбора. При включенном режиме происходит подсвечивание объектов сетки при наведении на них курсора мыши;
- реверс выделенных элементов;
- отбор элементов только на поверхности сетки;
- отображение идентификаторов отобранных элементов;
- вывод в окно информации дополнительных данных об отобранных элементах (например, об идентификаторе узла и его координатах);
- выбор типа отбираемых ячеек [2].

Рис. 2. Набор кнопок для управления режимами отбора элементов сеток

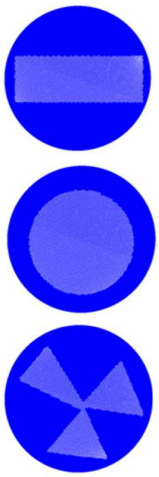

Рис. 3. Использование разных областей выделения

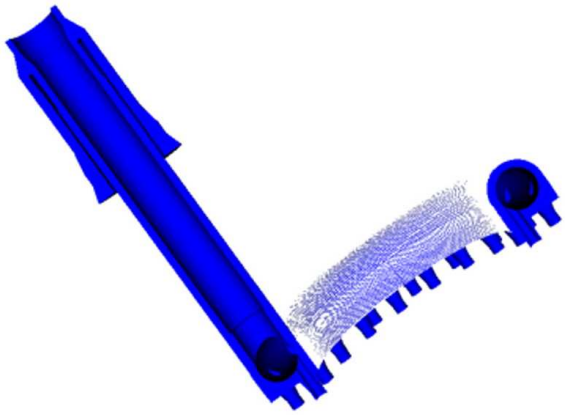

Рис. 4. Использование режима отбора элементов сетки по углу

Реализация отбора элементов с помощью фильтров VTK

В ППП ЛОГОС-Препост в качестве библиотеки для работы с трехмерными данными используется библиотека VTK [1]. До версии 4.0.6 эта же библиотека использовалась для реализации механизма отбора элементов. Обобщенная схема отбора с использованием фильтров VTK показана на рис. 5.

Рис. 5. Обобщенная схема отбора элементов с использованием фильтров VTK

В данной схеме весь механизм отбора можно условно разделить на три этапа:

1) формирование списка элементов, удовлетворяющих заданному условию. Этими элементами могут быть ячейки или узлы;
2) формирование из выбранных элементов необходимых пользователю данных (узлов, граней, ячеек, подобластей). Результатом этого этапа является VTK-сетка;
3) сохранение и передача VTK-сетки визуализатору. Сохранять сетку необходимо, так как результаты пикинга используются в других модулях ППП.

Представленная модель имеет ряд недостатков.

Во-первых, это большие затраты времени и памяти на выполнение процедур отбора. Причина в том, что промежуточными данными является VTK-сетка. Заметим также, что VTK-фильтры выполняются довольно медленно.

Во-вторых, в некоторых случаях VTK-фильтры отбирают элементы некорректно. Например, фильтры "не умеют работать" со свободными узлами, т. е. узлами, не принадлежащими ни одной ячейке.

В-третьих, VTK не поддерживает действия с гранями. Поэтому приходилось разрабатывать собственные фильтры или использовать конвейер фильтров для получения набора граней.

Все это отрицательно сказывалось на быстродействии и усложняло разработку новых операций отбора.

Реализация отбора элементов с использованием механизма операций

В версии 4.0.6 ППП ЛОГОС-Препост разработана новая модель отбора элементов (рис. 6). Рассмотрим ее основные особенности.

Во-первых, изменился способ хранения результатов отбора элементов. Если в предыду-

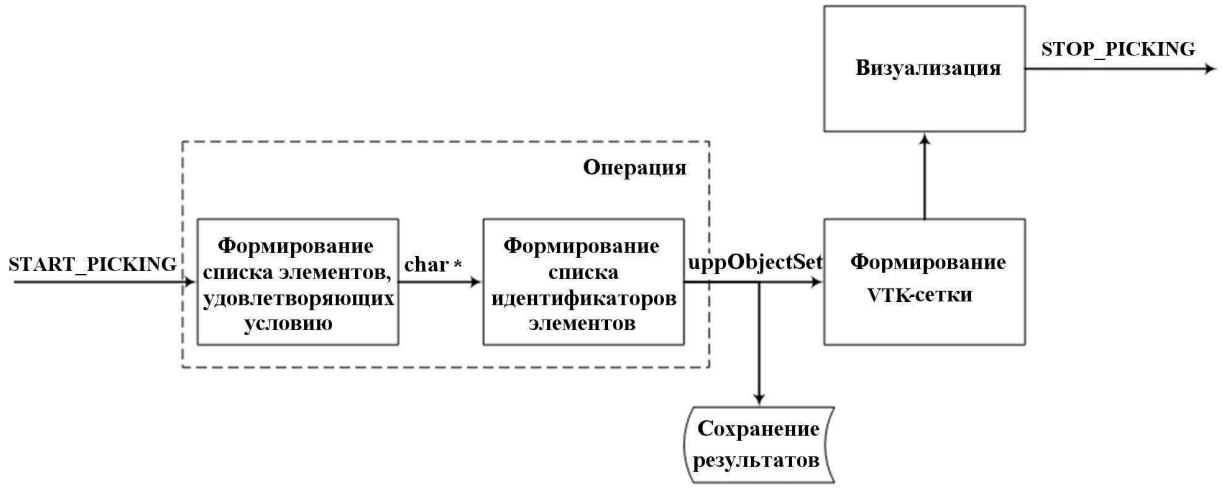

Рис. 6. Обобщенная схема отбора элементов с использованием механизма операций

щей модели сохранялась сетка, то в новой сохраняются только идентификаторы элементов. Это требует меньше памяти и ускоряет передачу данных между модулями программы.

Для преобразования идентификаторов отобранных элементов в VTK-сетку разработаны специальные функции формирования сетки:

GeneratePointsMesh — по списку идентификаторов узлов;

GenerateFacesMesh — по списку идентификаторов граней;

GenerateCellsMesh — по списку идентификаторов ячеек;

GenerateCellsMeshSurface — поверхностной сетки по списку идентификаторов ячеек.

Эти функции принимают на вход исходную сетку и идентификаторы отобранных элементов, а на выходе формируется сетка, которая может быть передана визуализатору для отображения на экране монитора.

Во-вторых, добавлена поддержка многосеточных данных, позволяющая загрузить несколько сеток одновременно и работать с ними в одном окне визуализации. Многосеточные данные имеют одну особенность: идентификаторы элементов, принадлежащие разным сеткам, могут совпадать. Поэтому был разработан специальный класс (`uppObjectSet`), предназначенный для связывания идентификатора сетки с идентификаторами элементов. В основе данного класса лежит вектор, хранящий пары *идентификатор сетки — массив списков идентификаторов элементов*.

В-третьих, VTK-фильтры были заменены операциями. С точки зрения бизнес-логики операция представляет собой законченную последовательность действий, которая оформлена в виде отдельного модуля [3]. На рис. 6 границы операции показаны пунктирной линией. Ниже дается описание всех доступных в данный момент операций.

SelectByArea. Предназначена для отбора элементов с помощью области выделения. Эта операция является одной из наиболее сложных в ППП.

На первом этапе формируются два массива `char`: список точек, лежащих в области выделения, и список точек, лежащих на границе области выделения. Для формирования списков используется класс `uppSelectionPickArea` и его методы `PointIsWithinArea` и `PointIsCrossingBoundary`. Методы принимают на вход координаты узла и возвращают значение `true`, если узел лежит внутри области выделения (`PointIsWithinArea`) или на границе области (`PointIsCrossingBoundary`), `false` — в противном случае.

На втором этапе для ячеек и граней выполняется дополнительная операция, идентичная для обоих типов. Рассмотрим ее на примере ячеек.

Формируется массив `cellsInArea`, размер которого равен числу ячеек в сетке. Каждое значение определяет, находится ли ячейка с данным идентификатором внутри области выделения (значение равно 0), вне этой области (значение 1) или пересекает ее (значение 2). Ячейка находится внутри области выделения, если все ее узлы внутри области выделения. Если все узлы вне области, то и ячейка вне ее. Иначе ячейка пересекает область выделения.

SelectByPoint. Предназначена для отбора одного элемента сетки. В данной операции в основном используются средства VTK. VTK-фильтр `vtkCellLocator` возвращает идентификатор ячейки, указанной пользователем. Для получения идентификатора узла выполняется цикл по всем узлам данной ячейки и выбирается узел, имеющий минимальное расстояние до точки, которую указал пользователь.

SelectElementsByAngle. Предназначена для отбора элементов сетки по углу между гранями.

На первом этапе при помощи VTK-фильтра `vtkCellLocator` выбирается поверхностная грань, на которую указал пользователь. Формируется список граней, удовлетворяющих условию: угол между нормалями к соседним граням меньше заданного. Первоначально для формирования списка поверхностных граней использовался VTK-фильтр `vtkConnectivityFilter`. Однако в связи с низким быстродействием он был заменен на класс `uppSelectionElementsByAngle`. Быстродействие удалось повысить и за счет того, что нормали стали вычисляться не для всей сетки, а только для соседних граней. Дополнительно использованы легкие статические массивы `char` для сохранения промежуточных данных.

На втором этапе выполняется проход по всем отобранным граням и сохранение идентификаторов элементов. При этом дубликаты автоматически удаляются.

SelectObjectElements. Предназначена для отбора элементов, принадлежащих выбранной подобласти. В ППП каждая подобласть визуализируется одним *актером*. В VTK актер является основным средством представления объекта на экране. Он "отвечает" за позицию, ориентацию и масштабирование данных, а также за визуальные свойства. Каждый актер связан с одной VTK-сеткой. Таким образом, цель данной операции — получить идентификаторы элементов, принадлежащие заданной сетке. По сравнению с `SelectElementsByAngle` в этой операции первый этап отсутствует, так как входными данными является сама сетка, а второй этап тот же самый.

SelectPointsByAngle. Предназначена для отбора узлов поверхностной сетки по углу между ребрами. В отличие от всех остальных эта операция не разбивается на этапы, так как выходными данными для нее являются только идентификаторы узлов.

Сначала при помощи `vtkCellLocator` выбирается указанная пользователем ячейка. Затем ищется ребро, концы которого имеют наименьшее растояние до точки, которую указал пользователь. Начиная с этого (*исходного*) ребра выполняется следующая процедура.

Рассматриваются все ребра ячеек поверхностной сетки, имеющие общую точку (узел) в одном из концов текущего ребра. Из них выбирается то ребро, которое образует с текущим наименьший угол, не превышающий заданного значения. После этого выбранное ребро становится текущим и процедура повторяется до тех пор, пока угол между ребрами не станет больше заданного.

Далее указанная процедура выполняется, начиная с другого конца исходного ребра.

При каждом выполнении процедуры концы текущего ребра становятся отбираемыми узлами поверхностной сетки.

Переход от VTK-фильтров к операциям позволил упростить расширение и модификацию функциональных возможностей ППП ЛОГОС-Препост.

Дополнительным преимуществом является то, что при выполнении операции происходит запись параметров отбора в файл сценария. Это дает возможность воспроизвести последовательность действий пользователя без использования визуализатора в режиме командной строки.

Отметим две особенности, которые позволили увеличить быстродействие механизма отбора. Во-первых, это использование параллельных вычислений в некоторых операциях. Во-вторых, изменение порядка визуализации. Дело в том, что сетка, сформированная в результате пикинга, больше нигде не используется. Поэтому при помощи установки соответствующего флага у актера, связанного с данной сеткой, она будет сразу же визуализироваться. Все остальные актеры при этом не обновляются. Это позволило значительно ускорить вывод данных на экран монитора для сеток больших размеров.

Исследование скорости отбора элементов

Были проведены тесты, сравнивающие скорость отбора элементов в ЛОГОС-Препост и в свободно распространяемых программах ParaView [4] и LS-PrePost [5]. Замер скорости отбора производился от момента выбора области выделения до отображения отобранных элементов включительно.

Сравнительные тесты проводились на компьютере со следующей конфигурацией:

- процессор Intel Core (TM) i5-2400 CPU 3.10 ГГц;
- ОЗУ 16.0 Гб;
- 64-разрядная операционная система.

В качестве исходной была взята сетка из файла *.vtk (рис. 7):

- размер исходного файла 1,5 Гб;
- количество узлов 17 268 546;
- количество поверхностных граней 4 135 824;
- количество ячеек 15 259 896.

Результаты сравнения полученных скоростей отбора элементов представлены на рис. 8.

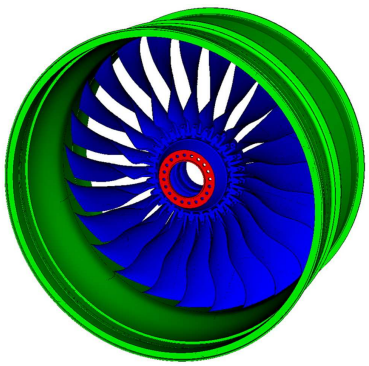

Рис. 7. Исходная сетка для тестирования

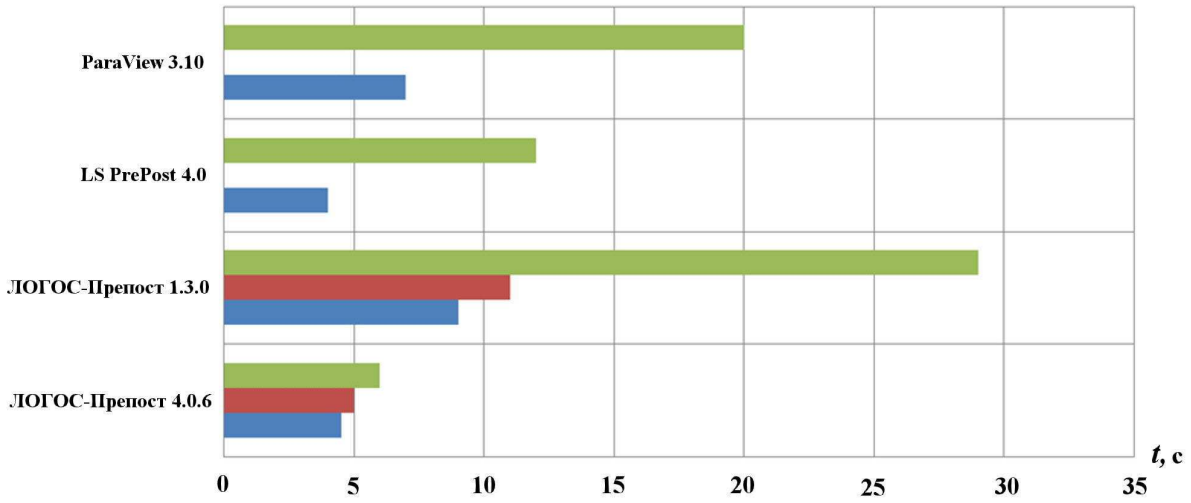

Рис. 8. Результаты сравнения скорости отбора элементов в программах-аналогах: сверху — ячейки; снизу — узлы; посередине — грани (только для ЛОГОС-Препост)

Так как LS-PrePost не поддерживает работу с сетками формата VTK, исходная сетка была преобразована средствами ППП в k-сетку.

ParaView тестировалась как в последовательном (использовалось одно ядро), так и параллельном режиме (использовались все доступные ядра процессора). Переключение режимов осуществлялось при помощи опции Use Multi-Core в настройках ParaView. Однако время отбора в параллельном режиме оказалось больше, поэтому в диаграмме сравнения на рис. 8 используются данные последовательного режима.

Заключение

Замена VTK-фильтров собственной реализацией позволила значительно уменьшить время отбора элементов сетки. Дополнительно удалось расширить функциональные возможности, предоставляемые библиотекой VTK (работа с гранями, многосеточными данными). Результаты тестов показывают, что по скорости отбора элементов на данный момент ЛОГОС-Препост превосходит зарубежные аналоги.

Следует, однако, заметить, что по функциональности пикинга, т. е. количеству доступных операций выбора элементов, ЛОГОС-Препост пока уступает своим конкурентам. Но переход на механизм операций позволяет довольно легко расширять функциональность новых его версий. Поэтому планируется продолжать развитие ППП в данном направлении.

Список литературы

1. VTK — The Visualization Toolkit. http://www.vtk.org.
2. *Гордеев А. В.* ЛОГОС-Препост. Особенности интерактивной работы с сеточными данными // Тр. XIV межд. конф. "Супервычисления и математическое моделирование", 1—5 октября 2012 г. Саров: РФЯЦ-ВНИИЭФ, 2013. С. 185.
3. *Анищенко А. А., Дерюгин В. И., Дюпин В. Н. и др.* Препостпроцессор ЛОГОС-Препост. Архитектура уровня бизнес-логики. Хранение, импорт и экспорт данных // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2014. Вып. 2. С. 78—82.
4. ParaView. http://www.paraview.org/.
5. LS-PrePost. http://www.lstc.com/lspp/.

Статья поступила в редакцию 22.07.13.