

12, декабрь 2015

УДК 004-1

**Автоматизация процессов перестроения программных реализаций
сложных вычислительных методов на основе графовых технологий
разработки программного обеспечения**

Колева А.С., студент

*Россия, 105005, г. Москва, МГТУ им. Н. Э. Баумана,
кафедра «Математика и компьютерные науки»*

***Научный руководитель:** Соколов А.П., доцент*

*Россия, 105005, г. Москва, МГТУ им. Н. Э. Баумана,
кафедра «Математика и компьютерные науки»*

bauman@bmstu.ru

Введение

В связи с возрастающим интересом к использованию новых композиционных материалов (КМ) в промышленности, что в особенности касается авиа- и ракетно-космической отраслей, резко возросло число практических задач, требующих анализа свойств КМ. К таким задачам следует отнести: задачи проектирования новых материалов с заранее заданными свойствами, анализа свойств КМ (упругих, теплофизических, прочностных и других), задачи сопряженного проектирования конструкции изделия с учетом используемых материалов при производстве.

На кафедре «Вычислительной математики и математической физики» МГТУ им. Н.Э. Баумана с 2004-го года ведутся работы по созданию Распределенной Вычислительной Системы GCD (PBC GCD). Основное назначение системы - решение задач инженерного анализа композитных конструкций с использованием методов компьютерного моделирования, базирующихся на математических моделях механики сплошных сред. В данной работе представлен базовый подход, примененный в последней версии (четвертой, v.4) программной реализации вычислительной подсистемы PBC GCD `gcdfes_Solver`.

В рамках решения различных инженерных задач механики композитных конструкций широко используется МКЭ. Очевидно, что возможность использования МКЭ

для целых классов различных задач влечет за собой необходимость выработки специального подхода при разработке каждой новой программной реализации численного метода решения конкретной задачи с его помощью. Наличие общего подхода обеспечивает минимизацию дублирования кода и, соответственно, повышение надежности программной реализации. С другой стороны, общий подход позволяет привлечь к разработке новых вычислительных схем, основанных на некотором общем методе (в частности, МКЭ) большее число специалистов.

С учетом неопределенности (невозможности заранее определить все практически значимые задачи, которые могут встать при проектировании новых композиционных материалов и конструкций на их основе в частности) было принято решение о создании автоматически реконфигурируемой реализации метода конечных элементов (РМКЭ), который лежит в основе метода асимптотического осреднения и методов многомасштабной асимптотической гомогенизации, которые лежат в основе численного решения задач механики композитов и конструкций на их основе.

В настоящей работе представлены основы программного подхода создания программных реализаций сложных вычислительных методов и представлена построенная сетевая модель программной реализации реконфигурируемого метода конечных элементов.

Теоретические основы и концептуальный подход

В процессе решения серии инженерных задач в области микромеханики композиционных материалов с помощью соответствующих специально разработанных конечно-элементных программных решений и др. естественным образом была построена иерархия классов на языке C++, обеспечивающая возможности решения ряда задач механики сплошной среды и в том числе некоторых прикладных задач (Рисунок 1). Следует отметить, что полученная иерархия классов, представленная в графическом виде (Рисунок 1) не обеспечивает прозрачное представление непосредственного процесса решения каждой конкретной задачи, что не облегчает процесс доработки новых вычислительных процедур.

При проведении анализа представленной диаграммы для решения каждой новой задачи теперь требуется создание нового подкласса – наследника наиболее подходящего из представленных. Данная иерархия классов является естественным результатом использования объектно-ориентированного подхода (ООП), используемого при проектировании программного обеспечения, основанного на «интерфейсах».

Известно, что наиболее оптимальные программные реализации вычислительных процедур создаются с применением структурного подхода (СП).

Этот подход существенно более прост для изучения, требует меньше опыта работы, тем самым позволяет инженеру-математику быстрее его освоить, однако, как результат, ведет к созданию алгоритмо - зависимого кода с последующими серьезными ограничениями по его сопровождению и доработке и, как следствие, существенной ограниченности использования полученных программных реализаций для классов практических задач.

Определение 1. Вычислительной операцией (ВО) будем называть алгоритм, обеспечивающий реализацию некоторого *численного метода*

Определение 2. Сложным вычислительным методом (СВМ) будем называть комплекс взаимосвязанных *вычислительных операций*, основанных на применении *численных методов* (возможно различных), по преобразованию входных данных, заданных численно, с целью получения численного результата.

Программные реализации **СВМ**, обеспечивающие решение частных исследовательских задач (далее ПО ЧЗ) сильно отличаются от программных реализаций, обеспечивающих решение практических задач (ПО ПЗ), в первую очередь предъявляемыми требованиями (см. табл.).

Требование	ПО для решения	
	частных исследовательских задач	практических задач
Валидация результатов	Не обязательна	Обязательна
Верификация результатов	Обязательна	Не обязательна
Возможно	Не обязательна	Обязательна
Возможность расширения	Часто не реализуется	Обязательна

Графовые представления СВМ

Как показывает практика: для программных реализаций вычислительных алгоритмов действительно эффективнее применять структурный подход, тогда как для более стандартных программных задач ввода-вывода данных - объектно-

ориентированный подход на основе «интерфейсов» или более современный на основе «стратегий».

Таким образом, для повышения эффективности создания программных реализаций СВМ необходимо было совместить преимущества объектно-ориентированного и структурного подходов.

Наиболее широко-используемым представлением алгоритмов являются блок-схемы.

Особенности использования блок-схем.

1. Блок-схемы удобны для описания небольших алгоритмов (число элементов не более 10-15).

2. Блок-схемы часто предполагают, что в их элементах описываются достаточно простые операции. В случае СВМ количество простейших операций астрономически велико, поэтому представление СВМ в виде одной блок-схемы нецелесообразно.

3. Блок-схемы не обеспечивают явного представления форматов, используемых данных, что для СВМ крайне важно.

4. Для блок-схем часто свойственно оперировать буквально скалярами, тогда как для СВМ операции производятся уже с векторами и массивами.

5. Блок-схемы крайне сложно «вкладывать» друг в друга. Другими словами: в связи с тем, что данные, над которыми работают функции в блок-схеме, элементарны и не имеют общей структуры для целого СВМ, любое вложение в отдельный блок блок-схемы другой блок-схемы приведет к необходимости предварительной подготовки данных в базовой блок-схеме и ее доработке. А в случае независимой разработке переработке каждой из блок-схем (и, естественно, соответствующих программных реализаций).

6. Блок-схемы не накладывают никакой смысловой нагрузки на ребра.

По этим причинам было принято решение использовать для описания процесса решения задачи с помощью СВМ понятие ориентированного графа.

С помощью ориентированного графа может быть представлен алгоритм решения вычислительной задачи и в том числе СВМ (Рисунок 1).

Представление в виде ориентированного графа (сетевой модели) обобщает представление алгоритмов в виде блок-схем и лишено всех указанных ранее недостатков.

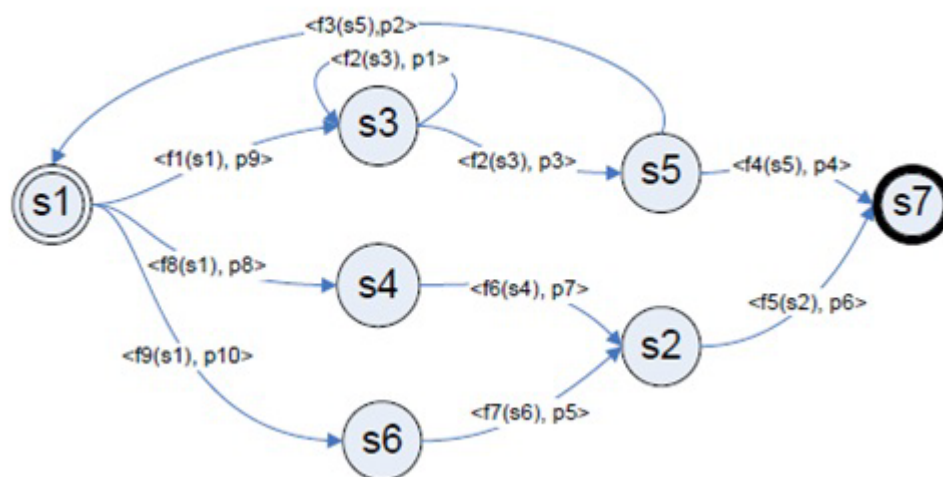


Рис. 1. Пример представления алгоритма в виде ориентированного графа

Определение 3. Данными CBM D будем называть фиксированное множество параметров (входных, вспомогательных и выходных) CBM, представляемых в виде скаляров, векторов или массивов значений.

Определение 4. Состоянием CBM S будем называть именованное фиксированное множество значений данных CBM.

Определение 5. Функцией перехода CBM F из состояния $S1$ в состояние $S2$ будем называть: $F(S1)=\langle f(S1), p \rangle = S2$,

где

$\langle f(S), p \rangle = p(S) / f(S): S$ – тернарный оператор языка C++;

$f: D \rightarrow D$ – функция-обработчик данных D в состоянии $S1$ и преобразующая их к состоянию $S2$;

$p: D \rightarrow B$ – функция-предикат, определяющая возможность перехода данных D из состояния $S1$ при помощи функции-обработчика f в состояние $S2$;

$B = \{0, 1\}$.

Определение 6. Сетевой моделью CBM (CM CBM) будем называть ориентированный граф $G=(S, E)$, где S – множество вершин, E – множество ребер, для которого:

- а) каждая вершина CM CBM определяет состояние CBM;
- б) каждое ребро CM CBM определяет функцию перехода из одного состояния в другое.

Пример некоторой CM CBM представлен выше (рис. 1).

Сетевая модель метода конечных элементов для решения задач статки механики сплошной среды

В основе вычислительной подсистемы Распределенной вычислительной системы GCD лежит метод асимптотического осреднения (MAO) В основе же MAO лежит применение метода конечных элементов MAO был модернизирован и развит в работах Программная реализация MAO была представлена в работе в составе созданной PBC GCD.

В связи с возрастающим интересом к использованию новых композиционных материалов (КМ) в промышленности, что в особенности касается авиа- и ракетно-космической отраслей, резко возросло число практических задач, требующих анализа свойств КМ К таким задачам следует отнести: задачи проектирования новых материалов с заранее заданными свойствами, анализа свойств КМ (упругих, теплофизических, прочностных и других), задачи сопряженного проектирования конструкции изделия с учетом используемых материалов при производстве.

Реальное применение метода стало возможным только после создания второй версии вычислительной подсистемы PBC GCD `gcdfes_Solver` в 2008-м году, которая обеспечила возможность автоматического перестроения программного вычислительного инструментария, обеспечивающего возможность проведения расчета в момент его старта. Во вторую версию вычислительной подсистемы PBC GCD вошла реконфигурируемая реализация МКЭ (РМКЭ), основанная на применении технологий DLL, плагинов и XML.

Однако и данные технологии не обеспечили удобных механизмов доработки созданных программных реализации, таким образом, была создана третья версия `gcdfes_Solver`, основанная на применении описанной выше графовой модели, что позволило:

1. обеспечить возможность фиксации метода решения классов частных практически значимых задач;
2. использовать независимые методы хранения создаваемых СМ СВМ (в PBC GCD специально была разработана реляционная структура данных, интегрированная в общую базу данных системы);
3. использовать уже созданные вычислительные подпрограммы (реализованные в виде функций, экспортируемых из DLL), определяя их в момент старта процесса расчета (свойство реконфигурируемости);

4. обеспечить возможность ведения совместной работы над СВМ коллективу специалистов, обеспечивая выполнение требования, представленных выше (см. табл. выше).

Используя представленный подход, была построена сетевая модель МКЭ (рис.2)

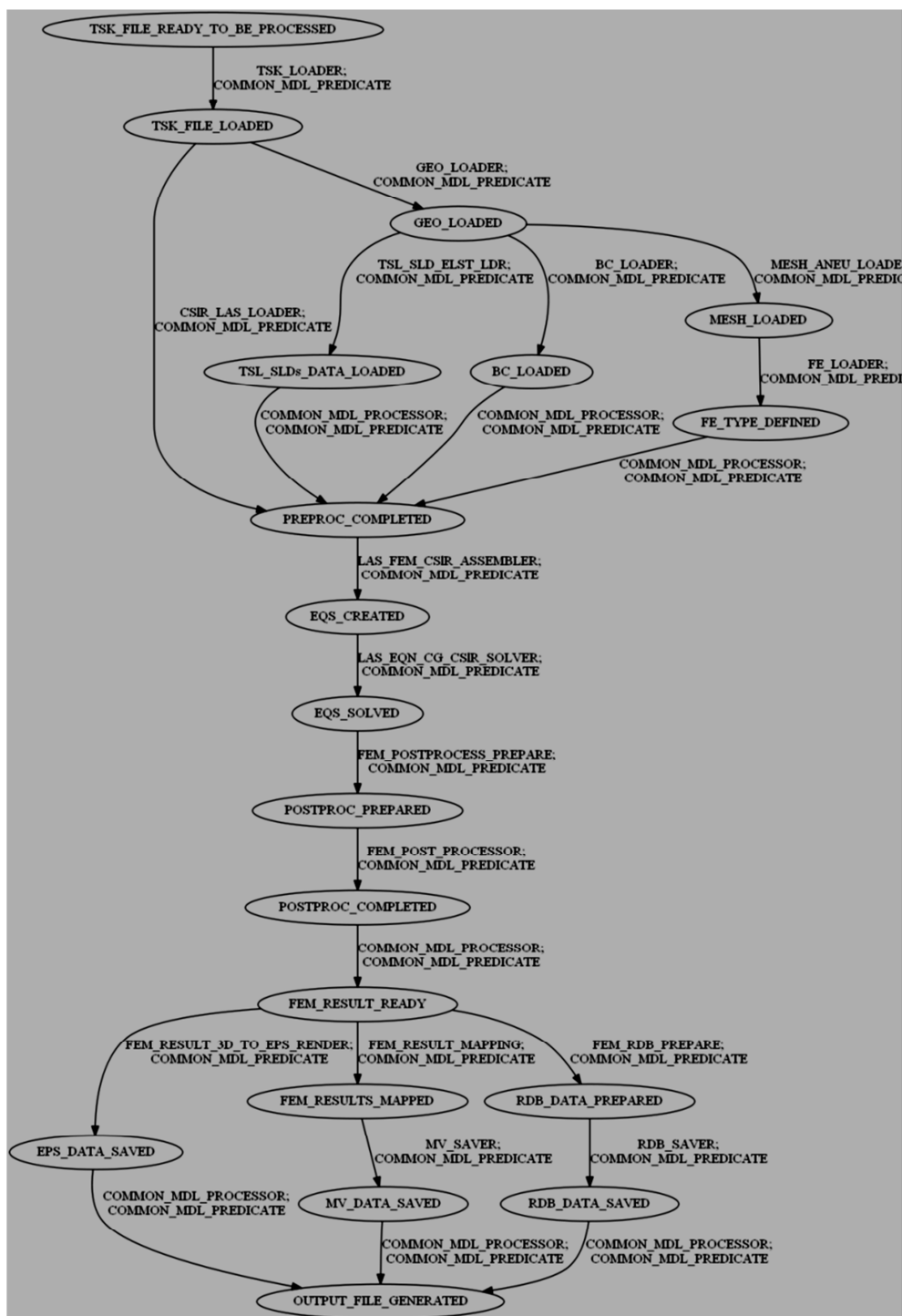


Рис. 2. Сетевая модель сложного вычислительного метода и метода конечных элементов

Выводы

1. Активное применение новой программной реализации подсистемы gcdfes_Solver обеспечило возможность решения ряда практических задач коллективом проекта в рамках различных математических моделей (упругости, прочности, теплопроводности, пластичности, термоупругости и пр.).

2. Предполагается, что использование новой методики создания СВМ позволит ускорить процесс создания программных реализаций СВМ при решении новых практических задач:

- нестационарные постановки механики сплошной среды;
- задачи устойчивости;
- задачи проектирования новых композиционных материалов с заранее заданными свойствами;
- решение задач оптимизации.

Список литературы

1. Оре О. Теория графов. М.: Наука, 1968. 336 с.
2. Кормен Т.М. Часть VI. Алгоритмы для работы с графами // Алгоритмы: построение и анализ. М.: Издательский дом Вильямс, 2006. 1296 с.
3. Митчел Э. Метод конечных элементов для уравнений с частными производными. М.: Мир, 1981. 216 с.
4. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ . 2-е изд. М.: Издательство БИНОМ, 1998. 560 с.
5. Дал У., Дейкстра Э., Хоор К. Структурное программирование. 1-е изд. М.: Мир, 1975. 247с.
6. Вержбицкий В.М. Численные методы. М.: Высшая школа, 2000. 267с.
7. Бахвалов Н.С., Панасенко Г.П. Осреднение процессов в периодических средах. Москва: Наука, 1984. 352 с.
8. Димитриенко Ю.И., Соколов А.П., Система автоматизированного прогнозирования свойств композиционных материалов // Информационные технологии. 2008. Вып. 1, №. 8. С. 31-38.