Главный редактор - д.э.н., профессор К.А. Кирсанов
тел. для справок: +7 (925) 853-04-57 (с 11^{00} – до 18^{00})
Опубликовать статью в журнале - http://publ.naukovedenie.ru

Ляпин Александр Александрович
Lyapin Alexander Alexandrovich
Заведующий кафедрой информационных систем в строительстве;
Head of the department of information systems in construction;

Панасюк Евгений Леонидович
Panasyuk Evgeny Leonidovich
Аспирант/postgraduate student
Ростовский государственный строительный университет
Rostov State University of Civil Engineering
05.23.17 – Строительная механика
E-Mail: johnphone@mail.ru

Использование видеокарт для выполнения вычислений при решении задач строительной механики методом конечных элементов

Finite elements method in structural mechanics using GPU systems

Аннотация: В данной работе описывается разработанная реализация метода конечных элементов с применением вычислительных возможностей видеокарт. Реализованы алгоритмы и структуры данных для параллельных вычислений в задачах строительной механики методом конечных элементов (МКЭ). Описаны основные аспекты программной реализации используемых алгоритмов на современных графических процессорах. Выполнено сравнение скорости решения задачи разработанным программным продуктом с одним из лидирующих программным МКЭ - комплексом ANSYS.

The Abstract: This work describes developed implementation of finite elements method with the use of general-purpose computing on graphics processing units. Parallel algorithms and data structures are implemented for problems of structural mechanics involving finite elements method. Described main aspects of implementation of used algorithms on modern graphics processing units. Fulfilled comparison of solution speed of developed software versus one of the leading FEM software package - ANSYS.

Ключевые слова: Метод конечных элементов, Параллельные вычисления, Графические ускорители, Метод сопряженных градиентов, Высокопроизводительные вычисления.

Keywords: Finite elements method, Parallel computations, General-purpose computing on graphics processing units, Conjugate gradient method, High performance computing.

***

Тактовая частота вычислительных устройств постоянно возрастала на протяжении последних шестидесяти лет. Но в настоящие время наблюдается спад темпов роста тактовой частоты. Одним из альтернативных путей повышения производительности является использование нескольких устройств для решения одной задачи - то есть параллельные вычисления.

Наряду с процессорами общего назначения, развивались используемые в графических ускорителях, предназначенные для расчётов трёхмерной графики реального времени - GPU (Graphic Processing Unit). Особенностью таких расчётов является независимость большого количества потоков данных (вершины и фрагменты графических примитивов можно обрабаты-

Главный редактор - д.э.н., профессор К.А. Кирсанов
тел. для справок: +7 (925) 853-04-57 (с 11^{00} – до 18^{00})
Опубликовать статью в журнале - http://publ.naukovedenie.ru

вать независимо), что позволяет эффективно использовать параллельные алгоритмы и параллельные вычислительные устройства. GPU состоят из большого количества узкоспециализированных ядер, которые работают параллельно. Так как в этих ядрах не требуется реализовывать весь набор возможностей CPU (Central Processing Unit), они занимают меньшую площадь микросхемы, что позволяет разместить множество ядер в одном процессоре. Движимые рынком с высокими потребностями пользователей к высококачественной трёхмерной графике, программируемые графические процессоры эволюционировали в высоко-параллельные, многоядерные вычислительные устройства с огромным вычислительным потенциалом и пропускной способностью памяти. Современные GPU могут выполнять порядка 30000 потоков параллельно, при этом общая вычислительная скорость современных видеокарт, измеряемая количеством операций с плавающей точкой в секунду (Floating point Operations Per Second, FLOPS), преодолела рубеж в один терафлопс.

Графические ускорители также имеют собственную оперативную память, пропускная способность которой в десятки раз превосходит стандартную.

Изначально видеокарты не позволяли использовать свои вычислительные мощности для расчётов, не связанных с графикой напрямую. Для расчётов данные к GPU передавались в формате изображений (текстур), а расчётные программы загружались как шейдеры. Недостатками такого метода являются сравнительно высокая сложность программирования, ограниченный набор возможностей взаимодействия нитей GPU и другие ограничения.

Вычисления на GPU развивались достаточно быстро и их производители, такие как NVIDIA и AMD, разработали свои инструментальные средства для использования видеокарт для неграфических расчётов. Можно выделить следующие средства программирования видеокарт: NVIDIA Cuda, AMD Stream, OpenCL.

Эффективность использования многоядерных процессоров зависит от применяемых алгоритмов. Существующие проекты по использованию видеокарт для расчётов показывают от двух до трёхсот кратного увеличения быстродействия по отношению к процессорам общего назначения.

Одной из основных задач строительной механики является нахождение внутренних усилий и деформаций конструкции от заданной внешней нагрузки. Для решения этой задачи на ЭВМ в большинстве случаев применяется метод конечных элементов (МКЭ) [1]. Это обусловлено эффективностью данного метода, а также относительной простотой реализации.

При расчётах конструкций методом конечных элементов большая часть времени уходит на составление и решение систем линейных уравнений. Уточнение расчётных схем вместе с усложнением самих объектов исследования приводит к необходимости существенного увеличения параметров конечно-элементной модели (количества конечных элементов, узлов в ансамбле). Это приводит к необходимости рассмотрения моделей, описываемых системами уравнений высокого порядка. Разумеется, понятие “системы высокого порядка” достаточно условно и зависит от уровня развития вычислительной техники [2]. В публикациях конца 50-х годов и начала 60-х годов к таковым относили системы примерно из пятнадцати – двадцати уравнений. Затем “планка” поднялась до ста, нескольких сотен, нескольких тысяч. Сегодня порог составляет от миллиона до нескольких десятков миллионов уравнений. Применительно к “суперкомпьютерам” типа “CRAY-Jaguar” можно говорить о сотнях миллиардах уравнений.

В данной работе описывается разработанная реализация метода конечных элементов с применением вычислительных возможностей видеокарт.

В качестве алгоритма решения СЛАУ использовался метод сопряжённых градиентов с предобуславливателем Якоби [3]. Данный метод обладает быстрой сходимостью, малой зависимостью от степени обусловленности матрицы, а также легко распараллеливается.

Главный редактор - д.э.н., профессор К.А. Кирсанов
тел. для справок: +7 (925) 853-04-57 (с 11^{00} – до 18^{00})
Опубликовать статью в журнале - http://publ.naukovedenie.ru

При реализации использовались: язык программирования C++ [4], среда разработки Microsoft Visual Studio 2008 Professional, система автоматизированной сборки CMake, выходной формат данных Xdmf, постпроцессор ParaView. Доступ к вычислительным возможностям видеокарт осуществлялся при помощи среды OpenCL.

Одним из параметров, от которых зависит время решения СЛАУ, является размерность задачи. Важно отметить, что программные реализации одного и того же метода решения СЛАУ, использующие различные аппаратные средства, имеют различный характер зависимости времени решения от размерности задачи. Поэтому сравнение скорости решения различных реализаций, необходимо проводить с использованием задач различной размерности.

Для возможности тестирования скорости решения на задачах разных размерностей, была выбрана модель, размерность которой задаётся несколькими параметрами.

Модель приближает совместную работу протяжённой плиты конечной жёсткости и упругого основания [5]. Усилия приложены к узлам верхней грани плиты.

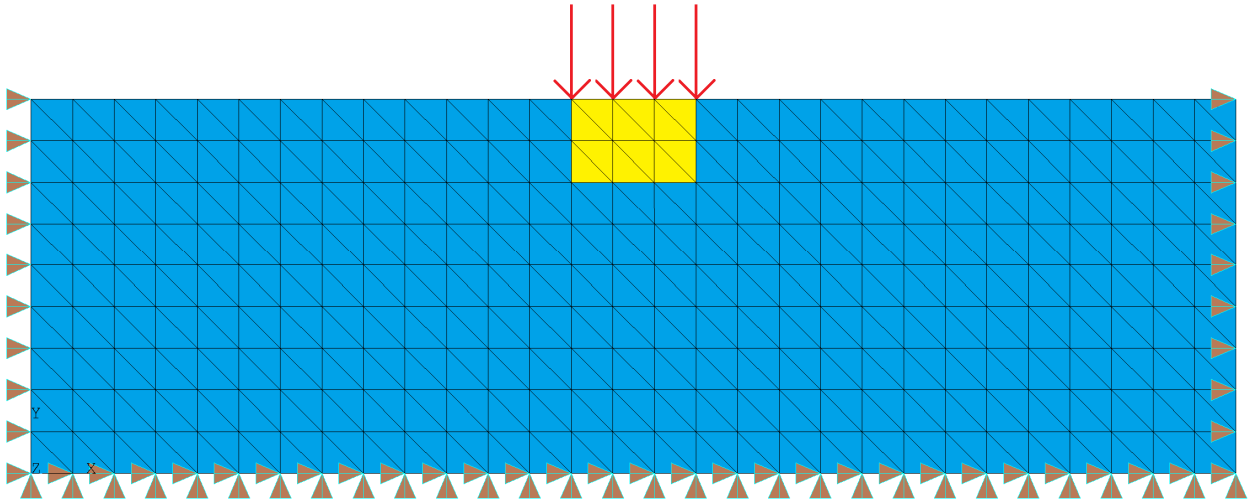

Рис. 1. *Условная схема расчётной модели малой размерности*

Параметрами данной модели являются количество узлов по вертикали и горизонтали. Данный набор параметров позволяет достаточно гибко менять как размерность задачи, так и ширину ленты системы уравнений. Хотя в итерационных методах ширина ленты не является важным параметром, однако он является критичным для прямых методов решения СЛАУ, эффективность которых резко снижается при увеличении ширины ленты.

Заметим, что на рис.1 приведена схема условно малого размера конечно-элементной аппроксимации. Реально решались задачи с количеством узлов от пятидесяти тысяч до полутора миллионов.

Для сравнения скорости работы разработанной программы использованы тестовые решения этих же задач с помощью программного комплекса ANSYS-12.1. При решении в среде ANSYS СЛАУ назначен тот же метод сопряженных градиентов с предобуславливанием по Якоби. Комплекс ANSYS, имеющий большую историю развития, известен тем, что большинство используемых им математических алгоритмов эффективно оптимизированы, и поэтому имеют высокую скорость работы. Т.е., сравнение скорости работы выполнялось с одной из лучших на сегодня реализаций алгоритма сопряженных градиентов на персональных компьютерах архитектуры x86-64. Использование одинаковых методов решения СЛАУ позволит достаточно объективно оценить эффективность разработанного программного обеспечения.

Главный редактор - д.э.н., профессор К.А. Кирсанов
тел. для справок: +7 (925) 853-04-57 (с 11^{00} – до 18^{00})
Опубликовать статью в журнале - http://publ.naukovedenie.ru

Решение задачи комплексом ANSYS выполнялось на высокопроизводительной современной персональной ЭВМ на базе процессора Intel Core I7 930. Данный процессор позволяет одновременно выполнять до восьми параллельных аппаратных потоков.

Решение системы СЛАУ комплексом ANSYS выполнялось параллельно с использованием восьми потоков, что обеспечивает наилучшую утилизацию ресурсов процессора CPU.

Количество итераций метода сопряженных градиентов в секунду использовалось в качестве одного из показателей скорости решения. Данный критерий позволяет оценить реальную скорость итерационного процесса решения СЛАУ. При использовании одинаковых алгоритмов общее число итераций тестируемой программы и ANSYS практически совпадает (некоторое расхождение возможно за счет разных алгоритмов округления результата в FPU и на арифметических процессорах видеокарты).

Программный комплекс ANSYS предоставляет подробную информацию о скорости выполнения различных этапов решения системы посредством файлов PCS.

Измерение скорости выполнения ядер в разработанном продукте выполнялось средствами предоставляемыми библиотекой OpenCL. Скорость выполнения итераций измерялась с помощью стандартных библиотечных функций C++.

Для контроля правильности решения производилось сравнение результатов.

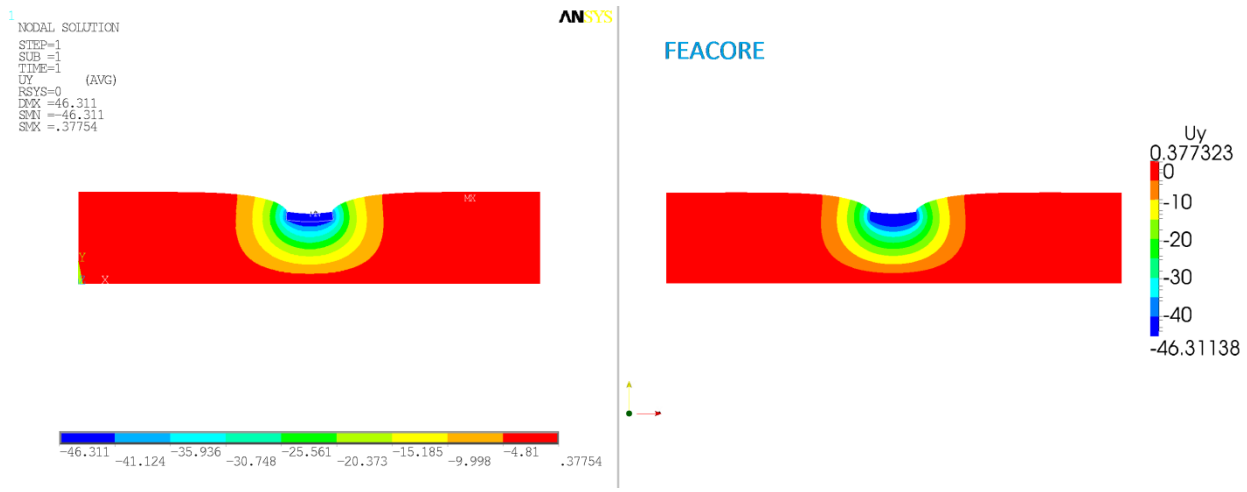

Рис. 2. *Сравнение полученных узловых решений для расчётной модели*

Результаты ANSYS и разработанной программы при одинаковой заданной точности достаточно близки - совпадают по крайней мере пять значащих цифр.

Измерение скоростных показателей разработанных алгоритмов проводилось на устройствах NVidia GTX 480 и NVidia Tesla C2050. Устройство Tesla C2050 поддерживает работу в режиме коррекции ошибок ECC (Error-Correcting Code), соответственно для данного устройства приводятся два варианта показателей.

Далее приводится график ускорения выполнения итераций относительно ANSYS по времени.

Главный редактор - д.э.н., профессор К.А. Кирсанов
тел. для справок: +7 (925) 853-04-57 (с 11^{00} – до 18^{00})
Опубликовать статью в журнале - http://publ.naukovedenie.ru

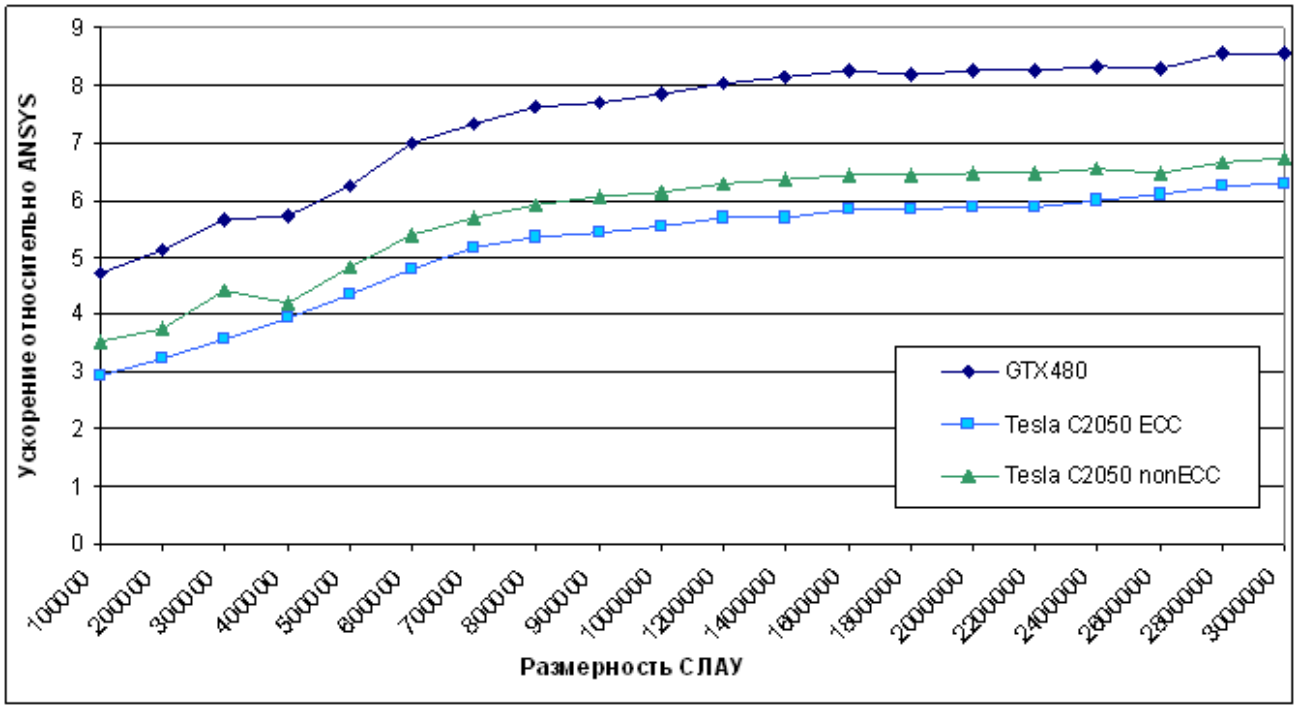

Рис. 3.*Ускорение итераций для расчётной модели*

Приведенные данные по относительным ускорениям (рис.3) полностью соответствуют с абсолютными результатами по времени решения задачи на FPU процессоров X86 в среде ANSYS и на видеокартах (рис.4).

Далее приведен график сравнения абсолютного времени решения СЛАУ (рис.4).

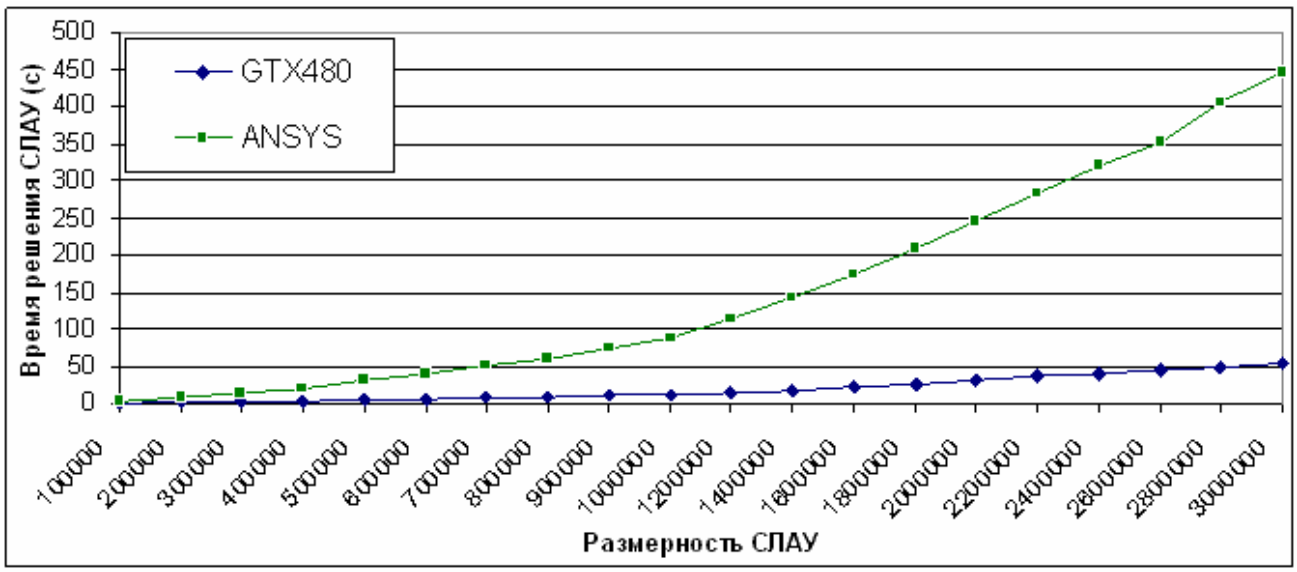

Рис. 4. *Время решения СЛАУ (с)*

Как видно, скорость разработанной программы на видеокарте практически на порядок быстрее, чем на CPU. С увеличением размерности задачи абсолютная разница времени значительно увеличивается, хотя отношение остается примерно постоянным.

Также было произведено сравнение скорости решения видеокарты и вычислительного кластера, находящегося в НИИ Механики и прикладной математики ЮФУ. Время решения

Главный редактор - д.э.н., профессор К.А. Кирсанов
тел. для справок: +7 (925) 853-04-57 (с 11^{00} – до 18^{00})
Опубликовать статью в журнале - http://publ.naukovedenie.ru

СЛАУ, состоящей из трех миллионов уравнений, на видеокарте NVidia GTX480 оказалось в 2 раза меньше, чем время решения с использованием двух стоек T-Edge 8.

Исследование показало, что при использовании быстрых многоядерных систем x86-64 вычисления на видеокарте имеют значительное преимущество по скорости (от 2 до 9 раз). Видеокарты могут быть эффективно использованы в решении задач строительной механики для систем высокой размерности. Использование алгоритма возможно во всех задачах строительной механики, в процессе решения которых многократно решаются СЛАУ. Это задачи нелинейной механики, динамики сооружений, ползучести, влагопереноса и теплопереноса, оптимизации конструкций.

ЛИТЕРАТУРА

1. Галлагер Р. Метод конечных элементов. Основы.- Москва: "МИР".- 1984.-С.428

2. Панасюк Л.Н. Прямые методы решения нестационарных задач теории сооружений. Диссертация на степень доктора технических наук по специальности 05.23.17 - строительная механика.- Ростов-на-Дону,: РГАС. - 1996. - С.389

3. Shewchuk J.R. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain. - Carnegie Mellon University Pittsburgh, PA, USA. - 1994.-С.64

4. Страуструп Б. Язык программирования С++. - Москва: "Бином-Пресс".- 2007.- С.57

5. Кадомцев М.И, Ляпин А.А., Селезнев М.Г. Исследование динамики заглубленных фундаментов методами граничных и конечных элементов // Строительная механика и расчет сооружений, № 3, 2010. С. 61-64.