**ОПЫТ ПРИМЕНЕНИЯ ГРАФИЧЕСКОГО УСКОРИТЕЛЯ К РЕШЕНИЮ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ В УПРУГОПЛАСТИЧЕСКОЙ ЗАДАЧЕ[[1]](#footnote-1)**

Халевицкий Ю.В., Коновалов А.В., Партин А.В.

Институт Машиноведения УрО РАН

# Введение

Математическое моделирование является одним из способов прогнозирования напряжений, возникающих в конструкциях и механизмах, как в процессе изготовления, так и в процессе их эксплуатации. Востребованным видом моделирования является прогнозирование формы и напряжённого состояния металла, возникающего при больших деформациях металла при высокой температуре. Необходимость подобных моделей возникает при проектировании и анализе технологических процессов обработки металлов давлением [1].

Наиболее требовательным к вычислительным ресурсам этапом решения подавляющего большинства задач математического моделирования является решение систем линейных алгебраических уравнений (СЛАУ) с большими разреженными матрицами. Выбор применяемых при решении такой системы алгоритмов и структур данных для хранения матрицы во многом зависит как от особенностей самой задачи, так и от используемого вычислительного оборудования. Необходимость адаптации реализаций алгоритмов для существующего оборудования подчёркивает актуальность исследований, посвящённых оптимизации производительности конкретных алгоритмов на реальной компьютерной платформе [2].

В последние годы широкое распространение получили графические вычислительные ускорители. Подобными ускорителями комплектуются как рабочие станции инженеров, так и специализированные вычислительные комплексы общего пользования. Применение графических ускорителей позволяет значительно ускорить решение задачи моделирования, однако перенос существующих программных комплексов, предназначенных для традиционных вычислительных устройств, на ускорители сопряжён со значительными трудностями. В данный момент поддержку графических ускорителей в вычислениях предоставляют лишь некоторые комплексы конечно-элементного анализа, такие, как ANSYS [3], Abaqus [4], LS-DYNA [5]. В силу коммерческой природы комплексов их авторы не раскрывают конкретные использующиеся модели и алгоритмы. Каждый из перечисленных конечно-элементных комплексов может быть использован для решения широкого круга задач моделирования. Неизвестно, являются ли применяемые алгоритмы оптимальными для использования при решении упругопластических задач. Известные авторам литературные источники не приводят сравнения подходов к решению СЛАУ, возникающих в упругопластических задачах.

В работе рассматривается применение некоторых распространённых итерационных методов решения СЛАУ, форматов хранения матриц, а также предобуславливателей к упругопластическим задачам.

# Модельная задача

Решение упругопластической задачи с помощью метода конечных элементов осуществляется в условиях пошагового нагружения. Каждый шаг нагружения состоит из трёх этапов [6]:

1. Расчёт локальных матриц жёсткости и векторов правой части для отдельных конечных элементов и сборка матрицы жёсткости *A*, применение граничных условий.
2. Решение системы линейных алгебраических уравнений
$$Ax=b$$
3. Вычисление напряжённо-деформированного состояния конечных элементов в конце шага нагружения

На каждом шаге нагружения этап 1 выполняется единожды, в то время, как этапы 2 и 3 повторяются от 10 до 15 раз для удовлетворения условию пластичности с приемлемой точностью. При этом матрица *A*, полученная на шаге 1, может быть использована повторно.

Исследование эффективности алгоритмов было выполнено на модельной задаче сжатия параллелепипеда плоскими плитами. Это позволило сравнивать полученные результаты вычислительных экспериментов с проведёнными ранее исследованиями, а также оценивать масштабируемость реализаций алгоритмов, получая сходные задачи с различным количеством конечных элементов.

Регулярная сетка параллелепипеда однозначно определяется числом разбиения *d*. Это число характеризует количество конечных элементов в направлении каждой из осей координат при рассечении параллелепипеда плоскостями, перпендикулярными данной оси.

Для оценки эффективности методов использовались сетки с большими разбиениями сравнимые с сетками, применёнными ранее при оценке эффективности итерационных методов на гомогенных кластерных системах [7].

# Вычислительное оборудование

Программное обеспечение для решения модельной задачи выполнялось на узлах гетерогенной части вычислителя кластерного типа «Уран», принадлежащего Институту математики и механики УрО РАН.

Каждый из узлов вычислителя, применявшийся в вычислительных экспериментах, оборудован 8 ускорителями Nvidia Tesla M2050, 48 гигабайтами оперативной памяти DDR3 и двумя шестиядерными процессорами Intel Xeon X5675, работающими на тактовой частоте 3.07 Гц. В данной работе при вычислениях было задействовано одно ядро процессора и один ускоритель.

Вычислительные мощности центрального процессора были задействованы для построения сетки модельной задачи, сборки матрицы жёсткости, а также для управления ускорителем в ходе решения системы линейных алгебраических уравнений.

# Использованные методы решения СЛАУ

В качестве набора реализаций алгоритмов решения СЛАУ была выбрана библиотека CUSP [9]. Некоторые из представленных в библиотеке итерационных алгоритмов, а также предобуславливателей могут быть использованы только для решения СЛАУ с положительно определёнными или симметричными матрицами жёсткости, поэтому при проведении вычислительных экспериментов были задействованы только подходящие алгоритмы и предобуславливатели.

Решение СЛАУ производилось до удовлетворения условия остановки $^{\left‖r\right‖}/\_{\left‖b\right‖}<10^{-9}$, где *r* – вектор невязки *b* – вектор правой части.

Сборка матрицы жёсткости производилась с использованием центрального процессора. Затем средствами библиотеки матрица жёсткости переводилась из координатного формата в соответствующий сжатый формат хранения. Перед решением СЛАУ матрица жёсткости копировалась в память ускорителя и применялся предобуславливатель.

Матрица в памяти ускорителя позже использовалась многократно для решения СЛАУ с различными правыми частями.

Библиотека CUSP предоставляет следующие форматы хранения матриц:

1. Гибридный формат (HYB).
2. Формат сжатой строки CSR (CSR).

Библиотека CUSP предоставляет два подходящих итерационных метода решения СЛАУ:

1. Метод биспоряжённых градиентов (BiCGSTAB) [9]
2. Метод обобщённых минимальных невязок (GMRES) [10]

Библиотека CUSP предоставляет два подходящих предобуславливателя:

1. Диагональный предобуславливатель (DIAG).
2. Предобуславливатель приближённой инверсии (AINV).

Без предобуславливателей оба метода сходились слишком медленно, и условие остановки не удовлетворялось после 1000 итераций.

При проведении вычислительных экспериментов проводились замеры астрономического времени, затраченного вычислительной системой на выполнение соответствующего участка вычислений. Инструментарий разработки Nvidia предоставляет более точные средства для анализа производительности кода, выполняющегося на ускорителе, привязанные к затраченным ускорителям тактовым циклам, однако измерение астрономического времени позволяет соотносить время работы ускорителя и центрального процессора, а также сравнивать производительность различных вычислительных систем.

На первом этапе вычислений, продолжительность которого обозначена в таблицах 1 и 2, как «Подготовка к решению» происходит преобразование матрицы из формата хранения, используемого при сборке в формат хранения, используемый при решении СЛАУ, а также копирование информации о матрице на ускоритель. Эти действия выполняются библиотекой прозрачно для программиста. На втором участке, обозначенном, как «Решение» происходит решение 15 различных СЛАУ с одинаковой матрицей жёсткости и различными правыми частями.

Было измерено время вычислений при разбиении $d=40$. Дополнительно для диагонального предобуславливателя были произведены замеры при разбиении $d=50$. Для каждой комбинации из метода хранения матрицы, числа разбиения, итерационного метода решения СЛАУ и предобуславливателя было произведено четыре замера времени, которые соответствуют последовательным применениям метода начальных деформаций на втором шаге нагружения.

# Результаты вычислительных экспериментов

## Решение СЛАУ с хранением матрицы в формате сжатой строки

В таблице 1 приводятся результаты вычислительных экспериментов при решении СЛАУ с хранением матрицы жёсткости в формате сжатой строки.

Таблица 1. Результаты вычислительных экспериментов при решении СЛАУ с хранением матрицы жёсткости в формате сжатой строки

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| d | Метод решения | Пред-обуславливатель | Подготовка к решению, с | Решение, с | Время решения СЛАУ, с |
| 40 | GMRES | AINV | 369,97 | 58,90 | 3,92 |
| 40 | GMRES | AINV | 370,00 | 57,99 | 3,87 |
| 40 | GMRES | AINV | 369,98 | 59,02 | 3,93 |
| 40 | GMRES | AINV | 370,03 | 57,97 | 3,86 |
| 40 | GMRES | DIAG | 13,96 | 43,05 | 2,87 |
| 40 | GMRES | DIAG | 13,97 | 42,92 | 2,86 |
| 40 | GMRES | DIAG | 13,96 | 43,03 | 2,84 |
| 40 | GMRES | DIAG | 13,06 | 45,03 | 3,00 |
| 50 | GMRES | DIAG | 26,05 | 107,94 | 7,19 |
| 50 | GMRES | DIAG | 26,05 | 107,10 | 7,14 |
| 50 | GMRES | DIAG | 26,06 | 107,88 | 7,19 |
| 50 | GMRES | DIAG | 27,04 | 108,78 | 7,25 |
| 40 | BICGSTAB | AINV | 370,92 | 104,93 | 6,99 |
| 40 | BICGSTAB | AINV | 369,99 | 96,98 | 6,46 |
| 40 | BICGSTAB | AINV | 369,04 | 102,99 | 6,87 |
| 40 | BICGSTAB | AINV | 368,94 | 110,02 | 7,33 |
| 40 | BICGSTAB | DIAG | 13,96 | 75,00 | 5,00 |
| 40 | BICGSTAB | DIAG | 13,96 | 75,04 | 5,00 |
| 40 | BICGSTAB | DIAG | 13,96 | 76,04 | 5,07 |
| 40 | BICGSTAB  | DIAG | 13,96 | 73,98 | 4,93 |
| 50 | BICGSTAB  | DIAG | 29,96 | 139,26 | 9,82 |
| 50 | BICGSTAB  | DIAG | 28,98 | 140,01 | 9,33 |
| 50 | BICGSTAB  | DIAG | 29,96 | 140,05 | 9,33 |
| 50 | BICGSTAB  | DIAG | 29,01 | 136,89 | 9,12 |

Из данных, приведённых в таблице 1 видно, что более сложная реализация AINV-предобуславливателя не только требует существенного времени для вычисления, но и не улучшает сходимость метода в сравнении с реализацией диагонального предобуславливателя. По всей видимости, это объясняется тем, что разработчики библиотеки стремились в первую очередь оптимизировать поставляемый AINV-предобуславливатель для симметричных матриц, а вариант предобуславливателя для несимметричных матриц формировался по остаточному принципу.

Простой в реализации на графических процессорах диагональный предобуславливатель позволяет добиться впечатляющего времени решения задач. При разбиении $d=40$ решение СЛАУ с использованием единственного ускорителя происходит быстрее, чем решение СЛАУ на нескольких узлах гомогенного кластера. Так, применение 8 гомогенных узлов (64 процессорных ядра) ранее позволило решить задачу методом обобщённых минимальных невязок за 3,4 секунды [7].

Сокращение времени решения задачи объясняется в первую очередь отсутствием необходимости в межпроцессорном взаимодействии.

Метод бисопряжённых градиентов показывает несколько худшее время решения, однако предыдущие исследования [7] показывают, что время вычислений методом бисопряжённых градиентов в меньшей степени подвержено деградации, вызванной межпроцессорным взаимодействием, что может быть важно при переходе к использованию нескольких ускорителей.

## Решение СЛАУ с хранением матрицы в гибридном формате

В таблице 2 приводятся результаты вычислительных экспериментов при решении СЛАУ с хранением матрицы жёсткости в гибридном формате

Таблица 2. Результаты вычислительных экспериментов при решении СЛАУ с хранением гибридном формате

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| d | Метод решения | Пред-обуславливатель | Подготовка к решению, с | Решение, с | Время решения СЛАУ, с |
| 40 | GMRES | AINV | 404,98 | 54,95 | 3,66 |
| 40 | GMRES | AINV | 403,99 | 54,00 | 3,60 |
| 40 | GMRES | AINV | 403,91 | 54,03 | 3,60 |
| 40 | GMRES | AINV | 403,94 | 54,05 | 3,60 |
| 40 | GMRES | DIAG | 25,01 | 39,03 | 2,60 |
| 40 | GMRES | DIAG | 25,00 | 38,10 | 2,54 |
| 40 | GMRES | DIAG | 25,00 | 38,10 | 2,54 |
| 40 | GMRES | DIAG | 25,00 | 38,99 | 2,60 |
| 50 | GMRES | DIAG | 50,07 | 82,07 | 5,47 |
| 50 | GMRES | DIAG | 50,07 | 81,87 | 5,45 |
| 50 | GMRES | DIAG | 50,96 | 80,99 | 5,40 |
| 50 | GMRES | DIAG | 50,98 | 84,87 | 5,66 |
| 40 | BICGSTAB | AINV | 401,96 | 100,92 | 6,73 |
| 40 | BICGSTAB | AINV | 401,92 | 106,92 | 7,13 |
| 40 | BICGSTAB | AINV | 400,04 | 95,11 | 6,34 |
| 40 | BICGSTAB | AINV | 400,96 | 92,12 | 6,14 |
| 40 | BICGSTAB | DIAG | 26,01 | 64,99 | 4,33 |
| 40 | BICGSTAB | DIAG | 26,01 | 65,96 | 4,39 |
| 40 | BICGSTAB | DIAG | 26,01 | 64,99 | 4,33 |
| 40 | BICGSTAB  | DIAG | 26,01 | 63,00 | 4,20 |
| 50 | BICGSTAB | DIAG | 49,00 | 135,98 | 9,06 |
| 50 | BICGSTAB | DIAG | 50,05 | 144,90 | 9,66 |
| 50 | BICGSTAB | DIAG | 55,98 | 137,96 | 9,19 |
| 50 | BICGSTAB | DIAG | 55,05 | 124,19 | 8,28 |

Сравнивая результаты, приведённые в таблицах 1 и 2 можно заметить, что гибридный формат показывает лучшее время вычислений в сравнении с форматом сжатой строки, однако соотношения затрат времени при решении различными методами и предобуславливателями идентичны полученным ранее результатам. Снижение времени решения достигается за счёт того, что большая часть строк матрицы *A* имеет одинаковое количество ненулевых элементов.

Преобразование матрицы в гибридный формат происходит медленнее, чем преобразование в формат сжатой строки. Это объясняется особенностями организации библиотеки CUSP.

# Выводы

Графические ускорители хорошо подходят для ускорения решения упругопластических задач. Использование единственного графического ускорителя позволяет добиться того времени решения задачи, для, которого ранее необходимо было задействовать небольшое количество узлов гомогенной кластерной системы.

В отличие от гомогенного вычислителя, метод обобщённых минимальных невязок является более предпочтительным при использовании гетерогенного вычислителя с единственным ускорителем.

Диагональный предобуславливатель является простым в реализации, но эффективным способом ускорить сходимость рассмотренных итерационных методов при решении модельной упругопластической задачи на графическом ускорителе.

В дальнейшем для решения задач с большим разбиением сетки следует задействовать в вычислениях несколько графических ускорителей, а также гетерогенный вычислитель кластерного типа. Требуется исследовать поведение различных форматов хранения матриц и предобуславливателей на задачах, отличных от модельной.

# Литература

1. *Коновалов А. В. Определяющие соотношения для упругопластической среды при больших пластических деформациях//Известия РАН. Механика твердого тела. 1997. № 5. С. 139-149.*
2. *Ильин В.П. Проблемы высокопроизводительных технологий решения больших разреженных СЛАУ // Параллельные вычислительные технологии (ПаВТ’2009): Труды международной научной конференции, Нижний Новгород, 30 марта – 3 апреля 2009 г. – Челябинск: Изд. ЮУрГУ, 2009. С. 166-175.*
3. *GPU Computing: [Электронный ресурс] // URL:* [*http://www.ansys.com/Products/Workflow+Technology/High-Performance+Computing/ANSYS+HPC+Packs/Features/GPU+Computing*](http://www.ansys.com/Products/Workflow%2BTechnology/High-Performance%2BComputing/ANSYS%2BHPC%2BPacks/Features/GPU%2BComputing) *(дата обращения: 17.04.2014)*
4. *NVIDIA GPUs Accelerate SIMULIA’s Abaqus FEA Solver : [Электронный ресурс] // URL:* [*http://www.nvidia.com/object/tesla-abaqus-accelerations.html*](http://www.nvidia.com/object/tesla-abaqus-accelerations.html) *(дата обращения: 17.04.2014)*
5. *Mr. Stan Posey and Dr. Srinivas Kodiyalam. Performance Benefits of NVIDIA GPUs for LS-DYNA // 8th European LS-DYNA Users Conference – Strasbourg – 23-24 May 2011.*
6. *Поздеев A. A., Трусов П.В., Няшин Ю.И. Большие упруго-пластические де-формации. М: Наука, 1986. 232 с.*
7. *А.В. Толмачев, А.В. Коновалов, А.С. Партин. Анализ эффективности ряда параллельных итерационных методов решения СЛАУ в упругопластической задаче на кластерной системе // Параллельные вычислительные технологии (ПаВТ'2013): труды международной научной конференции (1–5 апреля 2013 г., г. Челябинск). Челябинск: Издательский центр ЮУрГУ, 2013. С.* *545–550*
8. *Nathan Bell and Michael Garland. Cusp: Generic parallel algorithms for sparse matrix and graph computations, 2014. Version 0.3.1*
9. *H. A. van der Vorst, BI-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of non-symmetric linear systems, SIAM J. Sci. Stat. Comput. 13 (2) (1992) 631–644*
10. *Saad Y. Iterative methods for sparse linear systems, SIAM 2003. С. 447.*
1. Работа выполнена в рамках программы фундаментальных исследований Президиума РАН №15, проект 12-П-1-1025. [↑](#footnote-ref-1)