

# ПРИМЕНЕНИЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ В КОНТЕКСТЕ КОНЕЧНО-ЭЛЕМЕНТНОГО АНАЛИЗА ЗАДАЧ ГЕОМАГНИТНОГО ЗОНДИРОВАНИЯ

О.В. Шаповалов, В.В. Гетманский, В.С. Чалышев, Д.И. Крыжановский, С.Е. Ларин, П.С. Павлов

Настоящая работа посвящена применению конечно-элементного анализа для решения задач электростатики. Эта задача возникает при восстановлении структуры породы с помощью геомагнитного зондирования. Рассчитываются трёхмерные осесимметричные задачи с использованием конечных элементов первого и второго порядков. Для триангуляции используется бесплатная библиотека Triangle, для расчёта - библиотеки MKL, CUSP. Реализация метода выигрывает по скорости у пакета COMSOL при расчёте на аналогичной сетке. Максимальная погрешность решения при этом не превышает 1%.

## Введение

Одним из направлений современных геофизических исследований являются методы электрического каротажа, заключающиеся в изучении пород, непосредственно примыкающих к стволам скважин, по поведению в них электромагнитного поля. Обработка результатов каротажного зондирования в современных условиях невозможна без применения вычислительной техники и сложного программного обеспечения, использующего численные модели электромагнитного поля. В настоящее время на рынке программного обеспечения существуют коммерческие пакеты (например, ANSYS[1], COMSOL[2], Elmer[3]), которые подходят для решения подобных задач. Однако всё чаще задачи ставятся так, что получение их решения в стандартных пакетах за приемлемое время становится всё более и более проблематичным. Обусловлено это, прежде всего, ростом требований к качеству и скорости моделирования, вызванному бурным развитием высокопроизводительных вычислительных систем, наблюдающемуся в последние годы. При этом очевидно, что ввиду большой вычислительной ёмкости рассматриваемой задачи для её качественного решения необходимо задействовать все имеющиеся ресурсы, правильно распределив вычисления между процессорами и видеокартами.

Таким образом, в рамках данного исследования были поставлены следующие задачи:

- адаптация существующих алгоритмов и решений для параллельной реализации, работающей в гибридной вычислительной системе, использующей как процессоры, так и видеокарты;
- разработка программной библиотеки решения трёхмерных осесимметричных и неосесимметричных задач электростатики для гибридных высокопроизводительных вычислительных систем, поддерживающей совместимость с форматом моделей COMSOL.

## Решение прямой задачи электростатики методом конечных элементов

При исследовании горизонтально залегающих горных пород встаёт проблема определения местоположения и толщины пластов. Используемый здесь математический аппарат сводится к задаче многокритериальной оптимизации при заданном начальном приближении. Точная оценка проводится с использованием трёхмерной модели исследуемой области. Для предварительной оценки результатов зондирования достаточно использования осесимметричной модели исследуемой области и многовариантного расчёта, где варьируется толщина пластов горных пород.

В работе решалась прямая задача электростатики для нахождения распределения значений потенциалов в геофизическом разрезе. Математическая модель описывается с помощью неоднородного уравнения Пуассона, представленного ниже в цилиндрических координатах:

$$\frac{1}{r} \frac{\partial}{\partial r} \left( r \sigma(r, \varphi, z) \frac{\partial u}{\partial r} \right) + \frac{1}{r^2} \frac{\partial}{\partial \varphi} \left( \sigma(r, \varphi, z) \frac{\partial u}{\partial \varphi} \right) + \frac{\partial}{\partial z} \left( \sigma(r, \varphi, z) \frac{\partial u}{\partial z} \right) = f(r, \varphi, z) \quad (1)$$

где  $\sigma$  - диэлектрическая проницаемость,  $u$  - функция распределения потенциала,  $f(r, \varphi, z)$  - функция распределения зарядов.

При условии осесимметричности производная по  $\varphi$  будет равна нулю, и уравнение(1) принимает вид:

$$\frac{1}{r} \frac{\partial}{\partial r} \left( r \sigma(r, z) \frac{\partial u}{\partial r} \right) + \frac{\partial}{\partial z} \left( \sigma(r, z) \frac{\partial u}{\partial z} \right) = f(r, z) \quad (2)$$

В качестве метода дискретизации уравнения(2) был выбран хорошо зарекомендовавший себя для данных задач метод конечных элементов (КЭ)[4] с использованием треугольной адаптивной сетки. В предлагаемой реализации используются линейные или квадратичные треугольные КЭ.

Реализацию метода можно разделить на несколько этапов:

- задание расчётной схемы;
- триангуляцию расчётной области;
- расчёт методом конечных элементов;
- постпроцессинг.

### Задание расчётной схемы

Во время задания расчётной схемы происходит добавление геометрических объектов, назначение материалов, граничных условий и линий замеров. В качестве геометрических объектов доступны любые замкнутые и незамкнутые полигоны. Материалы назначаются на замкнутые контуры. Граничные условия могут быть назначены на линии и на замкнутые области. Замеры на данный момент могут проходить только по линии.

### Триангуляция

Для триангуляции использована бесплатная библиотека Triangle[5], работающая с методом триангуляции Делоне и выполняющая разбиение области в виде планарного графа с прямыми ребрами[6].

Хотя библиотека поддерживает генерацию сетки как линейных, так и квадратичных конечных элементов, в случае квадратичных КЭ генерация занимает очень большое время, около 40 - 50% от всего времени расчёта. Поэтому сетка всегда генерируется с линейными КЭ, а при расчёте с элементами второго порядка необходимые точки достраиваются уже без использования библиотеки Triangle. Данная оптимизация позволила сократить время триангуляции до 1 - 3% для сеток различных размерностей.

Для создания адаптивной сетки задаются ограничения по площади треугольников в замкнутых областях и минимальный угол треугольников. Для упрощения задания ограничений был написан алгоритм, позволяющий построить адаптивную сетку с заданным количеством треугольников. Он основывается на соотношении площадей замкнутых контуров области интегрирования.

### Расчёт методом конечных элементов

Расчёт методом конечных элементов состоит из следующих этапов:

- построение матрицы жёсткости для всех конечных элементов;
- ансамблирование глобальной матрицы жёсткости;
- решение системы линейных алгебраических уравнений (СЛАУ).

Матрица жёсткости для треугольного элемента при условии осесимметричности принимает вид

$$\left[ k^{(e)} \right] = \frac{\pi R \sigma}{2A} M \quad (3)$$

где  $\sigma$  -- диэлектрическая проницаемость,  $A$  - площадь треугольного элемента,  $R$  – расстояние от центра масс треугольника до оси симметрии.

Матрица  $M$  для элементов первого порядка имеет вид:

$$M = \begin{bmatrix} b_i b_i + c_i c_i & b_i b_j + c_i c_j & b_i b_k + c_i c_k \\ b_j b_i + c_j c_i & b_j b_j + c_j c_j & b_j b_k + c_j c_k \\ b_k b_i + c_k c_i & b_k b_j + c_k c_j & b_k b_k + c_k c_k \end{bmatrix} \quad (4)$$

где  $b$  и  $c$  - коэффициенты, зависящие от координат вершин треугольника следующим образом:

$$\begin{aligned} b_1 &= Y_2 - Y_3; & c_1 &= X_3 - X_2; \\ b_2 &= Y_1 - Y_3; & c_2 &= X_3 - X_1; \\ b_3 &= Y_1 - Y_2; & c_3 &= X_2 - X_1; \end{aligned} \quad (5)$$

где  $[[X_a Y_a.png]]$  - координаты  $a$ -ой вершины треугольника.

Матрица  $M$  для элементов второго порядка имеет вид

$$M = \begin{bmatrix} b_1^2 + c_1^2 & \frac{-b_1 b_2 - c_1 c_2}{3} & \frac{-b_1 b_3 - c_1 c_3}{3} & 0 & \frac{4(b_1 b_3 + c_1 c_3)}{3} & \frac{4(b_1 b_2 + c_1 c_2)}{3} \\ & b_2^2 + c_2^2 & \frac{-b_2 b_3 - c_2 c_3}{3} & \frac{4(b_3 b_2 + c_3 c_2)}{3} & 0 & \frac{4(b_1 b_2 + c_1 c_2)}{3} \\ & & b_3^2 + c_3^2 & \frac{4(b_2 b_3 - c_2 c_3)}{3} & \frac{4(b_1 b_3 + c_1 c_3)}{3} & 0 \\ & & & \frac{8(f(b_2, b_3) + f(c_2, c_3))}{3} & \frac{4(b_3 g_b + c_3 g_c + 2b_1 b_2 + 2c_1 c_2)}{3} & \frac{4(b_2 g_b + c_2 g_c + 2b_1 b_3 + 2c_1 c_3)}{3} \\ & & & & \frac{8(f(b_1, b_3) + f(c_1, c_3))}{3} & \frac{4(b_1 g_b + c_1 g_c + 2b_3 b_2 + 2c_3 c_2)}{3} \\ & & & & & \frac{8(f(b_1, b_2) + f(c_1, c_2))}{3} \end{bmatrix} \quad (6)$$

где  $f(p, q) = \text{png}$ .

Глобальная матрица жёсткости представляет собой сумму вкладов отдельных конечных элементов. Ансамблирование матрицы осуществляется простым суммированием матриц жёсткости отдельных конечных элементов по соответствующим вершинам.

### Постпроцессинг

Постпроцессинг заключается в заключительной обработке полученных на этапе вычислений результатов. Сейчас доступно сравнение с заранее сохранёнными расчётами в COMSOL, расчёт тока через выбранную поверхность, также возможно сохранение результатов в формат VTK для последующего просмотра в Paraview.

### Описание программной библиотеки

Разрабатываемое программное решение оформлено в виде библиотеки, позволяющей производить расчёты осесимметричных моделей пластов грунта методом конечных элементов по данным, снятым с вертикальных каротажных зондов. Библиотека состоит из фабрики объектов, на которой основан реестр их хранения. Реестр (рис. 1) реализует интерфейсы по работе с различными объектами, необходимым для расчёта модели (геометрические примитивы, материалы и граничные условия - для прямой задачи; варьируемые потенциалы и контролируемые параметры - для задачи динамической фокусировки).

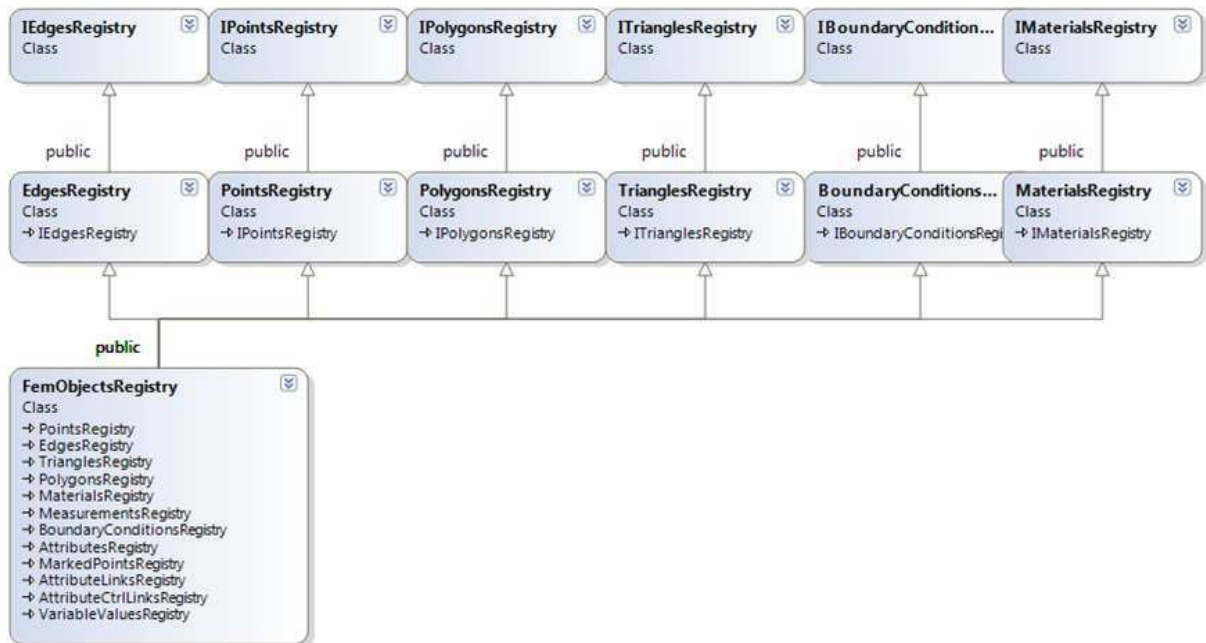


Рис. 1. Архитектура реестра библиотеки

Расчётная схема, включающая модель зонда и среды, задаётся полигонами. На полигоны назначаются материалы, граничные условия, поверхности замеров и т.д.

Для приведения геометрии в виде набора наложенных друг на друга полигонов к виду планарного графа написаны алгоритмы разбиения сторон многоугольников на части в случае их пересечения и наложения.

Уникальность точек обеспечивается заданным правилом сортировки по координатам. Для корректного упорядочения точек задействован дополнительный массив точек, определяемых с фиксированной точностью, соответствующей порядку погрешности сравнения вещественных чисел. Отсортированная коллекция

используется для быстрого поиска при добавлении новых точек, дабы исключать точки с одинаковыми координатами.

После триангуляции материалы, граничные условия и замеры назначаются треугольникам и точкам, то есть происходит поиск соответствия треугольников и точек геометрическим областям произвольной формы.

Изначально реализация препроцессинга занимала около 90% времени из-за использования медленных коллекций стандартной библиотеки C++ и алгоритмов линейного поиска. Для ускорения геометрических алгоритмов определения принадлежности точек и треугольников различным областям используется предварительное определение принадлежности этих элементов ограничивающему прямоугольнику для полигона. Алгоритмы поиска были заменены на бинарный поиск, а коллекции типа map на линейные массивы. За счет описанных преобразований время, затрачиваемое на препроцессинг сократилось до 30% на сетках относительно больших размерностей (от 10000 до 500000 треугольников). Таким образом, последовательное решение СЛАУ составляет около 70% от общего времени решения.

Отличительной особенностью разрабатываемой библиотеки является распараллеливание вычислений как на CPU, так и на GPGPU (General-purpose graphics processing units - техника использования графического процессора видеокарты для вычислений общего назначения) с использованием библиотек для параллельного решения СЛАУ с разреженными матрицами.

В качестве решателей СЛАУ с разреженной матрицей были использованы библиотеки MKL[7] и CUSP[8]. Первая использует многоядерный процессор, вторая – GPGPU технологию Nvidia CUDA.

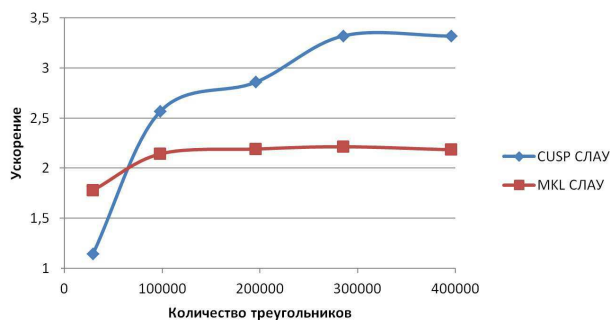
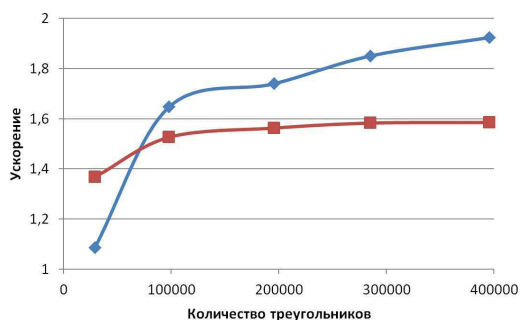
На этапе постпроцессинга вычисляются интегралы на основе посчитанных значений потенциалов в точках. Есть два способа расчёта: с использованием значений в узлах сетки и с использованием значений в произвольных узлах в области интегрирования. Первый способ требует более мелкой дискретизации области интегрирования, но работает быстрее. Это связано с тем, что во втором случае требуется выполнять интерполяцию значений для точек замера, то есть искать треугольник, к которому принадлежит точка, вычислять в ней значение потенциала и напряжённости электростатического поля. При добавлении расчёта напряжённости в узлах сетки в процессе решения и использовании только узлов сетки для расчёта значений интегралов время постпроцессинга составляет сотые доли процента от общего времени расчёта.

Для обеспечения совместимости с форматом модели COMSOL был разработан парсер, обеспечивающий распознавание основной функциональности создания модели из m-файлов модели COMSOL. Для реализации совместимости потребовалось создать аналогичную COMSOL по поведению логику создания модели, в частности, для разбиения рабочей области на полигоны используется основанный на использовании супервершин алгоритм поиска контуров. Также возникла необходимость обеспечения поддержки разных форматов задания модели, такой же как в COMSOL сортировки объектов, и большое количество других успешно решённых проблем. Все алгоритмы импорта были оптимизированы и суммарно составляют менее 0,1% процента от общего времени решения.

### Эксперимент

Тестирование производительности проводилось на стенде с Intel Core i5 2310 и GeForce GTX560. Замерялось ускорение решение СЛАУ и общее ускорение решения по сравнению с последовательным расчётом.

Полученные ускорения по сравнению с последовательным расчетом:



а) весь расчёт,

б) только решение СЛАУ}

По сравнению с пакетом COMSOL, который тоже использует параллельные библиотеки для решения СЛАУ, ускорение на CPU при расчёте с использованием разработанной библиотеки составляет от 1,5 до 2 раз. Для верификации результатов использовались данные, полученные при расчёте в пакете COMSOL v4.2. Для контроля точности была задана поверхность, через которую считался суммарный проходящий ток. Результаты приведены в таблице 1.

Таблица 1. Анализ погрешностей (eps) расчёта проходящего тока по сравнению с COMSOL, eps1 и eps2 - погрешности при расчёте с КЭ первого и второго порядка соответственно.

Количество треугольников	COMSOL	Последовательный расчёт	OpenMP
	30000	219	169,4
50000	527	305,7	106,1
100000	1377	712	232

На рисунке 2 показано сравнение расчёта распределения потенциала в расчетной области, полученных из COMSOL и с помощью разработанной библиотеки.

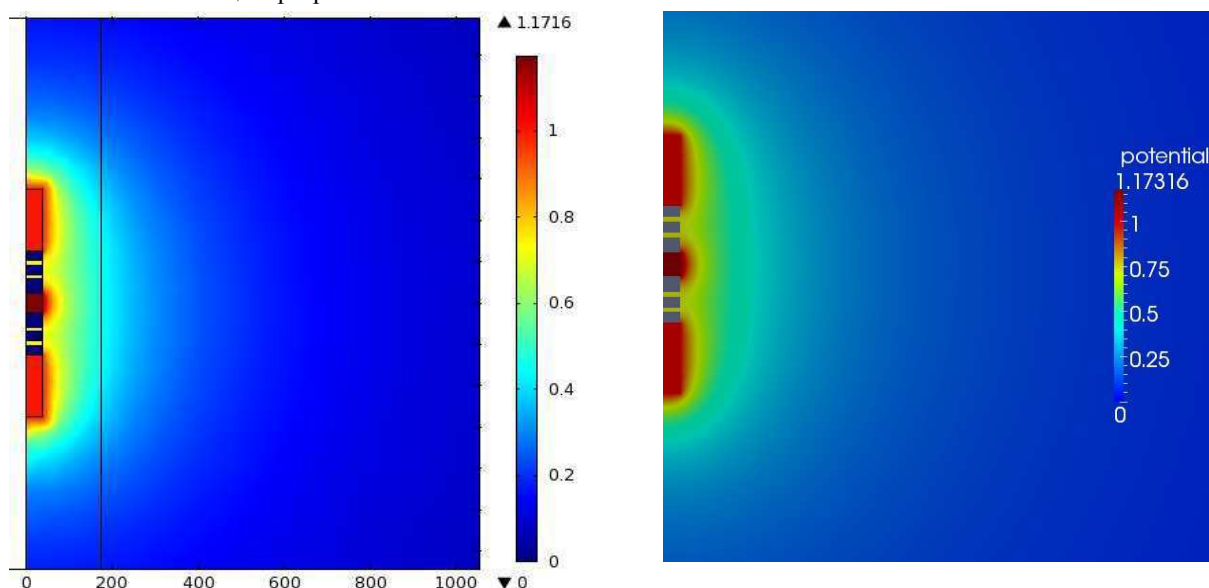


Рис.2. Расчёт потенциалов: а) с использованием пакета COMSOL, б) с использованием разработанной библиотеки}

В представленной модели использован метод динамической фокусировки для расчёта неизвестного значения потенциала в среднем электроде по заданным потенциалам на верхнем и нижнем электродах и условиях равенства потенциалов на средних электродах, разделённых изоляторами.

Как можно заметить, достигнутое ускорение при распараллеливании однократного расчёта далеко от идеального. Поскольку основным сценарием использования разработанной библиотеки является многовариантный расчёт, возможно эффективно распараллелить решение как для многопроцессорных машин, так и для кластера. Таким образом, было реализовано распараллеливание по различным вариантам расчёта с помощью технологий OpenMP. Используется равномерное распределение всех вариантов расчёта по процессорам системы. Тесты проводились на Intel Core i5 2310, результаты распараллеливания приведены в таблице 2.

Таблица 2. Ускорение многовариантного расчёта

Удельная проводимость среды	COMSOL	Расчёт с КЭ первого порядка	Расчёт с КЭ второго порядка	eps1	eps2
1,2	0,7441	0,741	0,734	0,0031	0,0101
3,4	1,6361	1,648	1,623	0,0119	0,0131
5,6	2,2508	2,278	2,238	0,0272	0,0128
7,8	2,7000	2,742	2,688	0,0420	0,0120
10	3,0426	3,098	3,031	0,0554	0,0116

### Заключение

Была разработана программная библиотека, ориентированная на численное решение задач геомагнитного зондирования. Она позволяет осуществлять расчёт прямой задачи электростатики, задачи динамической фокусировки, а также многовариантный расчёт. Модель может задаваться вручную или импортироваться из COMSOL. Имеющаяся реализация обеспечивает высокую точность и скорость расчёта. Распараллеливание вычислений производится с использованием оптимизированных под многопоточность библиотек MKL и CUSP, многовариантный расчёт распараллелен за счёт применения OpenMP и MPI.

### ЛИТЕРАТУРА:

1. ANSYS - Simulation Driven Product Development. - <http://www.ansys.com>
2. COMSOL Multiphysics Engineering Simulation Software. - <http://www.comsol.com>
3. Elmer Open Source Finite Element Software for Multiphysical Problems. - <http://www.csc.fi/english/pages/elmer>
4. Сегерлинд Л. Применение метода конечных элементов. - М: Мир, 1979. - 392 с.
5. A Two-Dimensional Quality Mesh Generator and Delaunay Triangulator. - <http://www.cs.cmu.edu/~quake/triangle.html>
6. Planar straight-line graph // Wikipedia. - [http://en.wikipedia.org/wiki/Planar\\_straight-line\\_graph](http://en.wikipedia.org/wiki/Planar_straight-line_graph)
7. Intel Math Kernel Library. - <http://software.intel.com/en-us/articles/intel-mkl>
8. Cusp-library - Generic Parallel Algorithms for Sparse Matrix and Graph Computations. - <http://code.google.com/p/cusp-library>