УДК 004.008

АНАЛИЗ ВОЗМОЖНОСТИ СОКРАЩЕНИЯ ОБЩЕГО ВРЕМЕНИ РАБОТЫ АЛГОРИТМА РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ МЕТОДОМ КОНЕЧНЫХ ЭЛЕМЕНТОВ

С.К. ДЕДЮЛИН,
Е.А. КАНУННИКОВА,
Н.И. КОРСУНОВ

Belgorod National Research University

e-mail:
d__sergey@list.ru
kanunnikova@bsu.edu.ru
korsunov@intbel.ru

В статье исследуется возможность ускорения вычислений при решении дифференциальных уравнений методом конечных элементов. Предлагается новый подход – алгоритм поэтапного рассмотрения области методом конечных элементов, главная идея которого заключается в том, чтобы рассматривать задачу, решаемую методом конечных элементов, не как одну большую задачу, а разделить ее на ряд более мелких подзадач. В этом случае снижение общего времени работы достигается не за счет снижения временной сложности алгоритма, а за счет использования меньших по объему входных данных.

В статье приведены результаты вычислительного эксперимента, целью которого являлось сравнение алгоритма метода конечных элементов и нового алгоритма поэтапного рассмотрения области методом конечных элементов. Критериями сравнения выступали общее время работы алгоритма и размер оперативной памяти необходимой для работы алгоритма.

Ключевые слова: метод конечных элементов, дифференциальные уравнения, временная и пространственная сложность алгоритма.

Решение многих физических задач может быть сведено к решению дифференциальных уравнений в частных производных. К распространенным задачам такого класса относится моделирование процессов теплопроводности, диффузии, конвекции, нестационарных полей различной природы, представляемые параболическими уравнениями, и стационарных полей (например, температурных или электромагнитных) представляемых эллиптическими уравнениями в частных производных. В большинстве случаев задача сводится к сверхбольшой системе линейных алгебраических уравнений, решение которой дает приближенное значение искомой функции в некотором наборе точек – узлов дискретизации.

На текущий момент существует два широко известных метода решений: метод конечных разностей и метод конечных элементов (МКЭ).

МКЭ сложнее в реализации метода конечных разностей. У МКЭ, однако, есть ряд преимуществ, проявляющихся на реальных задачах: произвольная форма обрабатываемой области; сетку можно сделать более редкой в тех местах, где особая точность не нужна. Полное описание метода конечных элементов приводиться в [5].

Рассмотрим этапы решения задач методом конечных элементов с общих позиций.

1. Выбор КЭ.

В одномерной задаче это просто отрезок прямой. В двумерной задаче – треугольный или четырёхугольный элемент. В общем случае любая фигура, с помощью которой можно разбить исследуемую область на непересекающиеся подобласти, но при этом следует учитывать, что чем сложнее элемент, тем с большими сложностями придётся столкнуться при вычислении интегралов, поэтому наиболее распространенными являются треугольный элемент и четырёхугольный со сторонами параллельными осям координат. Для трёхмерной области – тетраэдр и параллелепипед.

2. Разбиение области на КЭ.

В отличие от метода конечных разностей разбиение может быть неравномерным и априорно учитывать градиент фазовой переменной, то есть там, где предполагается быстрое изменение фазовой переменной, сетка должна быть гуще и наоборот. Существуют различные способы автоматического разбиения области на КЭ, поскольку исполнение этого этапа вручную утомительно и часто приводит к ошибкам.

3. Получение функции формы.

В зависимости от соотношения требований точности задачи и возможностей вычислительной системы выбирают линейную, квадратичную или более высокую степень функции формы, при этом число узлов аппроксимации должно быть минимум на единицу больше порядка аппроксимирующей функции.

4. Учёт граничных условий.
5. Получение матрицы жёсткости и вектора нагрузок.

На этом этапе используется метод взвешенных невязок в пределах одного КЭ. Суть метода взвешенных невязок заключается в следующем: подбирается функция, удовлетворяющая дифференциальным уравнениям и краевым условиям, но подбирается не произвольно, поскольку такой подбор вряд ли возможен уже в двумерном пространстве, а с использованием специальных методов.

6. Ансамблирование или получение глобальных матрицы жёсткости и вектора нагрузок.

Рассматривая поочередно каждый КЭ, располагаем элементы локальной матрицы жесткости в глобальной в соответствии с номерами улов подключения КЭ.

7. Решение системы алгебраических уравнений.

При решении может быть учтена особенность матрицы коэффициентов, поскольку она, как правило, имеет ленточную форму. Для решения получившейся системы алгебраических уравнений используем метод Жордана-Гаусса.

Недостатком МКЭ является быстрое возрастание времени работы алгоритма при увеличении размера исследуемой области, поскольку точность решения задачи обеспечивается использованием большого числа КЭ.

Использование суперкомпьютеров с общей или распределенной памятью требуют решения вопроса распараллеливания данной программы. В анализе последовательных программ на возможность их распараллеливания можно использовать несколько методов. В настоящее время находит широкое применение метод построения информационного графа программы и последующего его анализа, в результате которого выделяются циклы с независимыми итерациями [1].

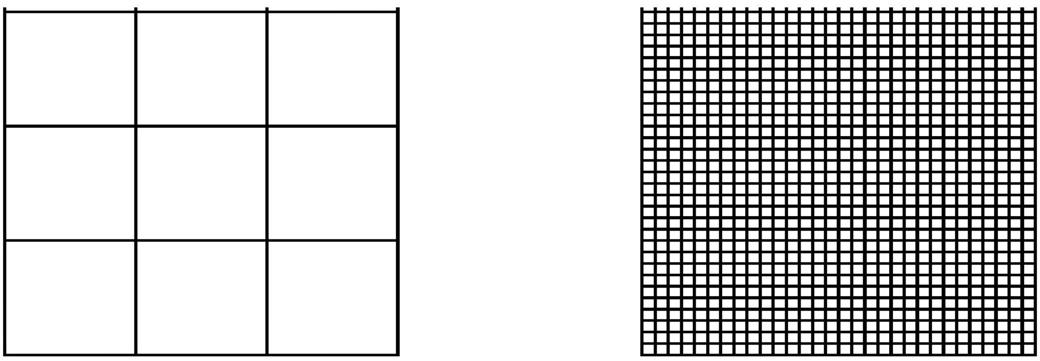
В соответствии с данным методом по тексту последовательной программы формируется циклический профиль, представляемый горизонтальными скобками и помеченной скобки, соответствующие независимым циклам. Каждый из циклов с внутренними операторами преобразователями, изменяющими значения переменных, ставят в соответствие вершинам информационного графа, ребра которого соответствуют условиям срабатывания операторов.

Выделив в информационном графе вершины, являющиеся итерационно-независимыми определяют возможность параллельного выполнения фрагментов последовательной программы. Однако, в [3] показано неэффективность данного метода.

Таким образом, повышение эффективности и распараллеливание вычислений при решении дифференциальных уравнений в частных производных методом конечных элементов требует разработки новых численных методов позволяющих использовать все возможности мощных суперкомпьютеров.

Ниже предлагается другой подход - алгоритм поэтапного рассмотрения области методом конечных элементов, то есть рассматривать задачу, решаемую методом конечных элементов, не как одну большую задачу, а разделить ее на ряд более мелких подзадач. Подобный подход описывался в работе [2] на примере использования метода инверсии и метода конечных разностей.

Допустим, исследуемая область имеет размеры 30x30 условных единиц, и требуется получить значение в каждом узле равномерной сети с шагом 1. На 1-ом этапе рассмотрим область 30x30 с шагом 3 (рис., *а*). На 2-ом этапе рассмотрим каждую из 9 подобластей размером 10x10, являющихся ячейками сети, полученной на предыдущем этапе, с шагом 1 (рис., *б*). Затем все данные объединяются.



а *б*

Рис. Схематичное изображение области 30x30:
а) 1 этап (шаг 10); *б*) 2 этап (шаг 1)

Алгоритм поэтапного рассмотрения области методом конечного элемента во много повторяет первоначальный алгоритм. Отличаются главным образом два этапа: разбиение области на КЭ (2-ой этап) и учёт граничных условий (4-ый этап), помимо этого, добавляется новый этап - сбор и объединение данных (8-ой этап).

8. Разбиение области на КЭ.

Исполнение этого этапа вручную утомительно, но автоматизировать данный процесс нелегко, так как он зависит от размера области и выбранного на каждом этапе шага. Необходимо выбрать максимально допустимое количество КЭ для области.

Проведем анализ трудоемкости алгоритма решения дифференциальных уравнений методом КЭ. Целью анализа трудоёмкости алгоритма является нахождение оптимального алгоритма для решения данной задачи. В качестве критерия оптимальности алгоритма выбирается трудоемкость алгоритма, понимаемая как количество элементарных операций, которые необходимо выполнить для решения задачи с помощью данного алгоритма. Функцией трудоемкости называется отношение, связывающие входные данные алгоритма с количеством элементарных операций.

Одним из упрощенных видов анализа, используемых на практике, является асимптотический анализ алгоритма. Целью асимптотического анализа является сравнение затрат времени и других ресурсов различными алгоритмами, предназначенными для решения одной и той же задачи, при больших объёмах входных данных. Используемая в асимптотическом анализе оценка функции трудоёмкости, называемая сложностью алгоритма, позволяет определить, как быстро растет трудоёмкость алгоритма с увеличением объёма данных.

Точное знание количества операций, выполненных алгоритмом, не играет существенной роли в анализе алгоритмов. Куда более важным оказывается скорость роста этого числа при возрастании объема входных данных. Она называется скоростью роста алгоритма. Небольшие объемы данных не столь интересны, как то, что происходит при возрастании этих объемов [4].

В асимптотическом анализе алгоритмов используются обозначения, принятые в математическом асимптотическом анализе.

Оценка *O* представляет собой верхнюю асимптотическую оценку трудоемкости алгоритма.

$$f(n) = O(g(n)), \tag{1}$$

где *f(n)* принадлежит классу функций, которые растут не быстрее, чем функция *g(n)* с точностью до постоянного множителя, *n* – размер входных данных.

Оценивая порядок сложности алгоритма, необходимо использовать только ту часть, которая возрастает быстрее других. Предположим, что рабочий цикл описывается выражением $N^3 + N$. В таком случае его сложность будет равна $O(N^3)$. Изучение быстро растущей части функции позволяет оценить поведение алгоритма при увеличении *N*. Например, при *N*=100, то разница между $N^3 + N$ =1000100 и N^3 =1000000 равна всего лишь 100, что составляет 0,01%. При вычислении O можно не учитывать постоянные множители в выражениях. Алгоритм с рабочим шагом $3N^3$ рассматривается, как $O(N^3)$. Это делает зависимость отношения $O(N)$ от изменения размера задачи более очевидной.

Вычислим асимптотическую оценку временной сложности алгоритма решения дифференциальных уравнений методом конечных элементов *f(n)*.

$$f(n) = O(g_1(n)) + O(g_2(n)) + O(g_3(n)), \tag{2}$$

где $g_1(n)$ сложность 5-го этапа алгоритма «Получение матрицы жёсткости и вектора нагрузок», $g_2(n)$ сложность 6-го этапа алгоритма «Ансамблирование или получение глобальных матрицы жёсткости и вектора нагрузок», $g_3(n)$ сложность 7-го этапа алгоритма «Решение системы алгебраических уравнений», *n* – количество КЭ.

$$f(n) = O(n) + O(n) + O(n^2 \log_2 n) = O(n^2 \log_2 n) \tag{3}$$

Так же, вычислим асимптотическую оценку пространственной сложности алгоритма решения дифференциальных уравнений методом КЭ $f_{om}(n)$. Асимптотическая оценка пространственной сложности алгоритма позволяет оценить необходимое количество оперативной памяти или дискового пространства, необходимой для работы алгоритма.

$$f_{om}(n) = O(3*n*8 + 9*n*8 + n^2*8 + 6*n*4) = O(n^2) \text{ байт}, \tag{4}$$

Отобразим данные об общем времени работы, полученные в ходе тестовых запусков приложения, и необходимом объеме оперативной памяти, полученные расчетным путем, для различного числа КЭ приведены в табл. 1.

Таблица 1

Данные о времени работы алгоритма и объеме оперативной памяти в зависимости от числа КЭ

Размер области, число КЭ	Общее время работы	Необходимый объем оперативной памяти
10x10=100	≈1 сек	≈100 Кб
40x40=1600	≈1 мин	≈20 Мб
100x100= 10000	≈4 часа	≈0,74 Гб
200x200=40000	-	≈11,92 Гб
1000x1000=1000000	-	≈7450,78 Гб

На основании данных таблицы 1, а именно общего времени работы алгоритма, следует выбрать максимальное число КЭ. Число КЭ стоит выбрать достаточно большим, для того чтобы было удобно рассматривать области и подобласти соответствующего размера, а время работы невелико. В качестве верхней границы числа КЭ стоит выбрать 1600-2000 КЭ. В качестве нижней границы числа КЭ стоит выбрать 100 КЭ.

В зависимости от выбранного максимального числа КЭ, на каждом из этапов алгоритма поэтапного рассмотрения области методом конечных элементов выбирается шаг. Количество этапов может варьироваться, в зависимости от размеров исходной области и выбранных значений шагов.

9. Учёт граничных условий.

На 1-ом этапе алгоритма поэтапного рассмотрения области методом конечных элементов учитываются граничные условия, указанные при постановке задачи. Стоит так же отметить, что возможна ситуация, когда граничные условия зафиксированы для точки, не совпадающей ни с одним из узлов сетки. В данной ситуации вместо регулярной сетки целесообразнее использовать нерегулярную.

В частности, для области 30x30 граничное условие зафиксировано для точки с координатами (27;20). Регулярная сеть для области 30x30 с шагом 10 задается с помощью 4 отметок на каждой из осей. Для учета точки с координатами (27;20) следует добавить по 1 отметки для каждой из осей, где точка не попала на уже имеющуюся отметку. В этом случае, сеть не будет регулярной, она будет образовываться отрезками - по оси X 5-ю отрезками, проходящими через отметки 0; 10; 20; 27 и 30,а по оси Y 4-мя, проходящими через отметки 0; 10; 20; 30.

На всех последующих этапах алгоритма поэтапного рассмотрения области методом конечных элементов учитывать первоначальные граничные условия не обязательно. Для каждой подобласти достаточно учитывать граничные условия только для 4-х значений в вершинах ячейки.

10. Сбор и объединение данных.

На данном этапе следует отметить одну важную деталь. Если рассматривать 2 соседние ячейки, например 1 и 2, то можно заметить ряд точек, общих для этих ячеек. Каждая из этих 2-х ячеек предложит свой вариант аппроксимирующего значения для данной точки. Причем оба значения будут отличаться от полученного классическим методом конечных элементов. Значение, полученное классическим методом конечных элементов, совпадает со среднеарифметическим значением 2-х значений с допустимой погрешностью (не выше 0,1%) от каждой из ячеек.

Для проведения вычислительного эксперимента, целью которого было сравнение последовательного алгоритма метода конечных элементов и алгоритма поэтапного рассмотрения области методом конечных элементов, была создана программная реализация этих алгоритмов. Критериями сравнения выступали общее время работы алгоритма и размер оперативной памяти необходимой для работы алгоритма.

В качестве объекта исследования была выбрана область 100x100 условных единиц. Требовалось узнать значение в каждом КЭ регулярной сети с шагом 1.

В последовательном алгоритме исследуемая область 100x100 рассматривается целиком с шагом 1.

В алгоритме последовательного рассмотрения области методом конечных элементов выделили 2 этапа. На 1-ом этапе выбрали шаг равный 10, получив регулярную сеть, разделяющую исходную область на 81 ячейку размером 10x10. На 2-ом этапе для каждой из 81 подобластей взяли шаг равный 1.

Результаты эксперимента приведены в табл. 2.

Таблица 2

Сравнение алгоритма метода конечных элементов и алгоритма последовательного рассмотрения области методом конечных элементов

Параметры	Алгоритм метода конечных элементов	Алгоритм последовательного рассмотрения области методом конечных элементов
Размер области	100x100	100×100
Число КЭ	10000	10000
Общее время работы	≈4 часа	≈8 сек
Необходимый объем оперативной памяти	≈0,74 Гб	≈100 Кб

Анализируя результаты эксперимента, стоит отметить, что общее время работы и пространственная сложность предложенного алгоритма значительно меньше. Отметим положительные стороны алгоритма поэтапного рассмотрения области методом конечных элементов.

1. На каждом из этапов алгоритма поэтапного рассмотрения области методом конечных элементов шаг внутри области может меняться, с помощью этого можно регулировать общее количество КЭ, или использовать на каждом из этапов фиксированный шаг, это избавит от необходимости постоянных расчетов локальных матриц жесткости. В каждом из описанных вариантов можно контролировать количество КЭ в рассматриваемой области либо с помощью изменений шага, либо с помощью изменения размера подобласти.
2. Другой немало важной особенностью является то, что после каждого из этапов алгоритма поэтапного рассмотрения области методом конечных элементов в наших руках будут находиться частичные итоговые данные, рассмотрение которых позволит принимать решения о необходимости детализации данных в данной подобласти, либо об отсутствии данной необходимости. Подобного рода отсев подобластей, может существенно сократить общее время работы.
3. Снижение общего времени работы достигается не за счет снижения временной сложности, а за счет использования алгоритма меньших по объему входных данных.
4. Снижение пространственной сложности.
5. Возможность высокоэффективного использования параллельных технологий. Это обуславливается тем, что расчеты в рамках этапов алгоритма поэтапного рассмотрения области методом конечных элементов для каждой из областей или подобластей происходят независимо от других. Тем самым не требуются дополнительные действия по синхронизации потоков или ожидания отстающих потоков. Передача данных между потоками требуется всего 2 раза: в начале нового этапа, когда рассылаются данные о граничных условия для каждой из подобластей, и в конце этапа, когда данные передаются для сохранения.

Отметим отрицательные стороны алгоритма.

1. Алгоритм поэтапного рассмотрения области методом конечных элементов сложнее в реализации.
2. Этап 2 «Разбиение области на КЭ» в автоматическом режиме производить намного сложнее.

На основании вышеизложенного можно сделать вывод, что новый алгоритм поэтапного рассмотрения области методом конечных элементов превосходит известный алгоритм метода конечных элементов по общему времени работы и пространственной сложности.

Список литературы

1. Воеводин, В.В. Параллельные вычисления / В.В. Воеводин, Вл.В. Воеводин. - Спб: Изд-во БХВ – Петербург, 2002. – 609 с.
2. Канунникова, Е. А. Математическое моделирование электрических полей методом инверсии: монография / Е. А. Канунникова. – Белгород: Изд-во БГТУ, 2010. – 92 с.
3. Корсунов, Н.И. Анализ распараллеливания вычислений при решений дифференциальных уравнений стационарных физических полей методом конечных элементов / Н.И. Корсунов, С.К. Дедюлин. Белгород: Изд-во Научные ведомости БелГУ, Серия: История. Политология. Экономика. Информатика, 2011 №13(108), Выпуск 19/1, 90-96 с.

4. Макконнелл, Дж. Основы современных алгоритмов 2-е дополненное издание / Дж. Макконнелл. Москва: Изд-во Техносфера, 2004. – 368с.
5. Сегерлинд Л. Применение метода конечных элементов / Сегерлинд Л. - М.: Мир, 1979. — 392 с.

ANALYSIS OF REDUGING ALGORITHM TOTAL TIME POSSIBILITY TO SOLVE DIFFERENTIAL EQUATIONS BY FINITE ELEMENT METHOD

S.K. DEDYULIN,
E.A. KANUNNIKOVA,
N.I. KORSUNOV

Belgorod National Research University

e-mail: d__sergey@list.ru
kanunnikova@bsu.edu.ru
korsunov@intbel.ru

In article calculations acceleration possibility is investigated at the solution of differential equations by a method of final elements. The new approach - algorithm of stage-by-stage consideration of area is offered by a method of the final elements, which main idea consists in considering the task solved by a method of final elements, not as one big task but to divide it into a number of smaller subtasks. In this case decrease in the general operating time is reached not at the expense of decrease in temporary complexity of algorithm, and at the expense of use smaller on volume of entrance data.

Results of the computing experiment which purpose was comparison of algorithm of a method of final elements and new algorithm of stage-by-stage consideration of area a method of final elements are given in article. As criteria of comparison the general operating time of algorithm and the size of random access memory necessary for algorithm work acted.

Keywords: finite element method, differential equations, the temporal and spatial complexity of the algorithm.