УДК 519.6

К вопросу о применении технологии OpenMP для численного решения интегральных уравнений

Тарасов Д.В., Лапкин В.В.

Пензенский государственный университет

Аннотация: В статье исследована целесообразность применения технологии параллельного программирования OpenMP для реализации различных способов численного решения интегральных уравнений. В качестве методов решения интегральных уравнений были выбраны методы имеющие разную вычислительную сложность: метод коллокации и метод конечных элементов. В работе представлены результаты замера времени при выполнении параллелизма как отдельных этапов решения (вычисление интегралов по квадратурным формулам, решения системы линейных алгебраических уравнений), так и для задачи в целом.

Ключевые слова: технология OpenMP, интегральные уравнения, параллельные вычисления.

1. Введение

Реализация численных методов нередко требует задействовать достаточно большое количество ресурсов ЭВМ, что негативно сказывается на времени выполнения программы. Если же реализуемый алгоритм не требователен к объему оперативной памяти, а предполагает лишь выполнения большого числа арифметических операций, то в этой ситуации удобно использовать многопроцессорные вычислительные системы с общей памятью. На сегодня под данные параметры подходят даже домашние ПК, оснащённые многоядерными процессорами, в которых каждое из ядер представляет собой независимое функционирующее вычислительное устройство.

В настоящее время существует много технологий которые позволяют достаточно легко организовать параллельные вычисления, одна из самых популярных на текущей момент это OpenMP – открытый стандарт для распараллеливания программ на языках C, C++ и Фортран. Даёт описание совокупности директив компилятора, библиотечных процедур и переменных окружения, которые предназначены для программирования многопоточных приложений на многопроцессорных системах с общей памятью [1].

2. Применимость технологии OpenMP для решения интегрального уравнения

К разработке параллельной программы, реализующей решение интегральных уравнений, стоит приступать только после написания последовательной программы для однопроцессорной вычислительной системы. И затем уже, в случае, когда время выполнения или другие вычислительные ресурсы оказываются приоритетными, следует переходить к параллельным вычислениям. Применение OpenMP на языках C/C++ сводится к добавлению в уже в написанную программу директив процессора, которые обрабатываются до начала компиляции программы.

Конечно, распараллелить можно не все алгоритмы, а лишь те, которые содержат многопотоковые участки, например, циклы.

Решение интегральных уравнений численными методами предполагает сведение последних к аппроксимирующей системе линейных алгебраических уравнений (СЛАУ). При этом нахождение элементов расширенной матрицы этой СЛАУ, как правило, требует нахождение интегралов или сумм. Таким образом, можно выделить следующие этапы для последующей параллельной реализации:

1) вычислительная схема решения интегрального уравнения, приводящая к СЛАУ. В частности, вычисление интегралов или сумм, составляющих коэффициенты данной СЛАУ;

2) решение аппроксимирующей СЛАУ.

Все представленные этапы решения в программной реализации будут содержать циклы, а значит и могут быть подвергнуть параллелизму. Рассмотрим все указанные этапы на примере решения интегрального уравнения методом коллокаций (как более простой вычислительной схеме) и методом конечных элементов (который содержит заметно большее число вычислений), и проанализируем временные затраты при использовании библиотеки OpenMP.

Этап «Вычисление коэффициентов аппроксимирующей СЛАУ». В методе коллокации коэффициенты СЛАУ определяются как значения определенных интегралов. Подчеркнем, что если они не является особыми, то в большинстве случаев для его вычисления достаточно применения формулы прямоугольников. Параллельная реализация формулы прямоугольников с применение технологии OpenMP к нахождению коэффициентов аппроксимирующей СЛАУ (например, в методе коллокаций) представлена на рис. 1 [2], а результаты замера времени – в табл. 1.

```
float rect_integral(double (*f) (double), float a, float b, int n) {
        float h = fabs((b - a) / n);
        float sum = 0;
        #pragma omp parallel reduction (+: sum)
        {
        # pragma omp for
                for (int i = 0; i < n; ++i)
                        sum += f(a + i * h) * h;
        }
        return sum;
}
```

Рис. 1. Параллельная реализация метода прямоугольников с использованием технологии OpenMP

Для обозначения параллельных участков кода используется директива `#pragma omp parallel`. Директива `#pragma omp for` служит для распределения итерации цикла между потоками; параметр директивы `shared` сообщает потокам что переменные, описанные данной директивой являются общими для всех потоков. Кроме того, параллелизм достигается за счет коллективной операции «+» и происходит при помощи параметра `reduction` директивы `for`.

В случае, когда вычисление коэффициентов аппроксимирующей СЛАУ сводится к вычислению двойных интегралов (например, решение интегральных уравнений методом конечных элементов), то их можно вычислить с помощью метода Симпсона. Для этого область интегрирования делится на элементарные прямоугольники. А параллелизм будем строить по циклу, выполняющему суммирование всех значений

Таблица 1. Время вычисления интеграла с помощью формулы прямоугольников

Узлы разбиения n	Время расчёта на одном ядре, с	Время расчёта на двух ядрах, с
10	0.00015	0.0011
50	0.014	0.032
100	0.093	0.1
500	9.21	5.23
1000	73.35	54.39

по каждому такому элементарному прямоугольнику, где работает формула:

$$\int_a^b \int_c^d f(x,y)dxdy \approx \frac{(b-a)(d-c)}{9}\left(f(a,c)+f(a,d)+f(b,c)+f(b,d)+4f\left[\left(a,\frac{c+d}{2}\right)+\right.\right.$$
$$\left.\left.+f\left(b,\frac{c+d}{2}\right)+f\left(\frac{a+b}{2},c\right)+f\left(\frac{a+b}{2},d\right)\right]+16f\left(\frac{a+b}{2},\frac{c+d}{2}\right)\right). \quad (1)$$

Результаты замера времени параллельных вычислений в зависимости от число узлов разбиения n на подобласти (в которой используется локальная формула Симпсона) по каждой из осей показаны в табл. 2.

Таблица 2. Время выполнения метода Симпсона для вычисления двойных интегралов в обычном режиме и при использовании технологии OpenMP

Узлы разбиения n	Время расчёта на одном ядре, с	Время расчёта на двух ядрах, с
10	0.26	0.48
30	4.89	4.54
50	20.3	16.28
100	156.6	78.2
150	460.2	312.76

Этап «Решение аппроксимирующей СЛАУ». Рассмотрим ещё один этап решения нашей задачи, а именно, решение системы линейных алгебраических уравнений большой размерности, которую так же возможно распараллелить, так как в данных алгоритмах присутствуют циклы.

В табл. 3 приведено время решения системы линейных алгебраических уравнений методом Якоби в случае использования технологии OpenMP и без нее при различных значениях размерности системы. Число потоков было выбрано равным количеству ядер процессора, в нашем случае два [3].

Таблица 3. Время решения системы линейных алгебраических уравнений

Размерность СЛАУ	Время расчёта на одном ядре, с	Время расчёта на двух ядрах, с
100×100	0.000873	0.00121
500×500	0.0214	0.0242
1000×1000	0.0915	0.0853
5000×5000	2.173	1.228
10000×10000	9.254	4.420

Предпочтение методу Якоби отдано поскольку аппроксимирующая матрица A нашей СЛАУ, полученная после применения метода коллокаций или метода конечных элементов, зачастую будет иметь вид близкий к диагональному (с сильно разряженной матрицей). А для корректной работы метода Якоби как раз и необходимо, чтобы диагональные элементы матрицы A были ненулевыми [4].

Также анализировались и другие численные методы решения СЛАУ. Например, исследователь может остановиться и на классическом методе Гаусса. Однако данный метод показал существенно большее общее время расчета [2].

3. Анализ результатов

Метод коллокации. Имея результаты распараллеливания частей кусков кода, можно объединить наработки параллелизма и применить их разом. В табл. 4 представлены результатами времени выполнения (с использованием OpenMP и без) метода коллокации [5, с. 543–550] для модельного уравнения:

$$x(t) + \int_0^{2\pi} \cos(t)\cos(\tau)x(\tau)d\tau = \sin(t),$$

где $x(t) = \sin t$ – точное решение.

Таблица 4. Время решения интегрального уравнения методом коллокации

Узлы коллокации	Время расчёта на одном ядре, с	Время расчёта на двух ядрах, с
10	0.0003	0.0013
50	0.017	0.029
100	0.1	0.18
500	11.25	6.69
1000	93.36	59.21

Из табл. 4 видно, что до определённого момента мы не только не получаем ускорение, а скорее замедляем выполнение программы, хотя и не существенное, так как речь идёт о долях секунд. Но когда задача становиться более объёмной и требовательной к ресурсам ЭВМ, параллельный код выполняется куда быстрее кода, написанного для одного процессорного ядра.

Метод конечных элементов. Аналогично рассмотрим результаты параллелизма метода конечных элементов. В табл. 5 приведены результаты замера времени выполнения метода конечных элементов [6, 7], с одновременным применением параллелизма для обычных и двойных интегралов [3]. В качестве модельного примера было выбрано уравнение:

$$\int_0^1 (t^3 + \tau^2)x(\tau)dt = -2\cos(t^2) + \cos(t^2+1) + 2\sin(t^2+1),$$

где $x(t) = \sin t$ – точное решение.

Как видно из табл. 4–5, использование параллельных конструкций с помощью директив OpenMP может привести к увеличению производительности программы, особенно при работе с большими объемами данных. Однако во всех случаях исследователю необходимо правильно выбирать параметры директив и балансировать нагрузку между потоками, чтобы избежать блокировок или переключений контекста, которые могут отрицательно сказаться на производительности.

Таблица 5. Время выполнения метода конечных элементов в обычном режиме и при использовании технологии OpenMP

Размерность СЛАУ	Время расчёта на одном ядре, с	Время расчёта на двух ядрах, с
5	0.63	0.25
15	11.51	2.47
30	67.51	21.48
50	313.36	112.9

4. Заключение

В статье исследована целесообразность распараллеливания программного кода с использованием технологии OpenMP при численном решении интегральных уравнений. Если программная реализация численного метода требует значительных временных ресурсов, то распараллеливание кода актуально. В этом случае применение технологии OpenMP достаточно просто. Так даже на двухядерной машине время решения интегрального уравнения методом конечных элементов было заметно снижено уже при размерности аппроксимирующей СЛАУ равной 50, для метода коллокации – 500. Однако параллелизм решения СЛАУ следует внедрять для заметно больших размерностей матрицы.

Литература

1. OpenMP Application Programming Interface. Version 6.0 Preview 1, November 2022. URL: https://www.openmp.org/wp-content/uploads/openmp-TR11.pdf

2. Тарасов Д. В., Лапкин В. В. Использование технологии OpenMP на многопроцессорных вычислительных системах с общей памятью для решения интегральных уравнений // Аналитические и численные методы моделирования естественно-научных и социальных проблем : сб. ст. по материалам XVII Всерос. с междунар. участием науч.-техн. конф. (г. Пенза, Россия, 28 ноября – 3 декабря 2022 г.) / под ред. д-ра физ.-мат. наук, проф. И. В. Бойкова. Пенза : Изд-во ПГУ, 2022. С. 150–154.

3. Лапкин В. В., Тарасов Д. В. Применение технологии OpenMP к численному решению интегральных уравнений // Математическое и компьютерное моделирование естественно-научных и социальных проблем : сб. ст. по материалам XVII Всерос. с междунар. участием науч.-техн. конф. молодых специалистов, аспирантов и студентов (г. Пенза, Россия, 1–4 июня 2023 г.) / под ред. д.ф.-м.н., проф. И. В. Бойкова. Пенза : Изд-во ПГУ, 2023. (в печати)

4. Вержбицкий В. М. Численные методы. Линейная алгебра и нелинейные уравнения : учеб. пособие для вузов. М. : Высшая школа, 2000. 266 с.

5. Канторович Л. В., Акилов Г. П. Функциональный анализ. М : Главная редакция физико-математической литературы, 3-е изд., 1984. 752 с.

6. Сегерлинд, Л. Применение метода конечных элементов: пер. с англ. М. : Мир, 1979. 392 с.

7. Тарасов Д. В. Решение интегральных уравнений теории линейных антенн методом конечных элементов // Журнал Средневолжского математического общества. 2023. Т. 25, № 1. С. 554–564. DOI: https://doi.org/10.15507/2079-6900.25.202301.554-564

MSC 65Y05

On the application of OpenMP technology for the numerical solution of integral equations

D.V. Tarasov, V.V. Lapkin

Penza State University

Abstract: The expediency of parallelization of program code using OpenMP technology for numerical solution of integral equations is studied. Methods with different computational complexity were chosen as methods for solving integral equations: the collocation method and the finite element method. The paper presents the results of measuring the time when parallelism is performed both for individual stages of the solution (calculation of integrals by quadrature formulas, solving a system of linear algebraic equations) and for the problem as a whole.

Keywords: OpenMP technology, integral equations, parallel computing.

References

1. OpenMP Application Programming Interface. Version 6.0 Preview 1, November 2022. URL: https://www.openmp.org/wp-content/uploads/openmp-TR11.pdf

2. Tarasov D. V., Lapkin V. V. Using OpenMP technology on multiprocessor computing systems with shared memory to solve integral equations. Analiticheskie i chislennye metody modelirovaniya estestvenno-nauchnykh i sotsial'nykh problem: sb. st. po materialam XVII Vseros. s mezhd. uchastiem nauch.-tekhn. konf. (g. Penza, Rossiya, 28 noyabrya – 3 dekabrya 2022 g.) [Analytical and numerical methods for modeling natural science and social problems: proceedings of the 17th All-Russian scientific and engineering conference with international participation], Penza, 2022. P. 150–154. (In Russ.)

3. Lapkin V. V., Tarasov D. V. Application of OpenMP technology to numerical solution of integral equations. Matematicheskoe i komp'yuternoe modelirovanie estestvenno-nauchnykh i sotsial'nykh problem : sb. st. po materialam XVII Vseros. s mezhdunar. uchastiem nauch.-tekhn. konf. (g. Penza, Rossiya, 1–4 iyunya 2023 g.) [Mathematical and computer modeling of natural science and social problems: proceedings of the 17th All-Russian scientific and engineering conference with international participation], Penza, 2023. (in print) (In Russ.)

4. Verzhbitsky V. M. Numerical methods. Lineynaya algebra i nelineynye uravneniya : ucheb. posobie dlya vuzov. [Linear algebra and nonlinear equations : textbook. manual for universities]. Moscow: Vysshaya shkola, 2000, 266 p. (In Russ.)

5. Kantorovich L. V., Akilov G. P. Funktsional'nyy analiz [Functional analysis]. Moscow: Glavnaya redaktsiya fiziko-matematicheskoy literatury, 3rd ed., 1984, 752 p. (In Russ.)

6. Segerlind L. Primenenie metoda konechnykh elementov: per. s angl. [Application of the finite element method: trans. from English], Moscow: Mir, 1979, 392 p. (In Russ.)

7. Tarasov D. V. Solution of integral equations of linear antenna theory by finite element method. Zhurnal Srednevolzhskogo matematicheskogo obshchestva. 2023. 25. 1. P. 554-564. (In Russ.) DOI: https://doi.org/10.15507/2079-6900.25.202301.554-564