УДК 621.396

ПРИМЕНЕНИЕ МЕТОДОВ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ ПРИ ПРОЕКТИРОВАНИИ ЭЛЕКТРОДИНАМИЧЕСКИХ ОБЪЕКТОВ

© 2016 Г. В. Воронцов

Воронежский институт высоких технологий

В работе рассматривается алгоритм оценки характеристик рассеяния электродинамических объектов на основе метода интегральных уравнений. Указано, что для ускорения вычислений могут быть использованы механизмы распараллеливания.

Ключевые слова: связь, электродинамический объект, параллельные вычисления.

В классическом подходе решения задач, которые связаны с излучением и рассеянием электромагнитных волн, например, в средствах связи, для получения точных вычислений используются аналитические методы. Но, так как такие решения существуют только для ограниченного класса задач, часто, для их решения ищут приближенное решение [1-4].

Появление же быстродействующих вычислительных машин дало новые возможности в решении широкого круга задач. И больше всего используют метод интегрального уравнения, так как он наиболее удобен для реализации на ЭВМ при решении задач, связанных с излучением и рассеянием электромагнитных волн [5, 6].

Например, чтобы найти распределение тока в антенне, возбуждаемой от заданного источника, задача сводится к решению неоднородного интегрального уравнения. И на ЭВМ такое интегральное уравнение может быть решено численно после преобразования его к матричному уравнению. А уже решение матричного уравнения (в виде того же СЛАУ) можно распределить на множество потоков, или процессоров, если есть многопроцессорная система. Конечно, никто не мешает распределить и между процессорами, и между потоками в каждом процессоре решение задачи [7-10]. А зная численное решение интегрального уравнения, достаточно легко получить распределение тока в антенне, входной импеданс, диаграмму направленности и т. д.

Также следует сказать, что ЭВМ позволяет не только относительно быстро решить задачу и найти приближенное численное решение интегрального уравнения, но и найти решение задачи с любой указанной точностью за относительно небольшое время, что избавляет от использования аналитических методов в особо сложных задачах [11-13].

Как было сказано выше, чтобы решить интегральное уравнение, его необходимо привести к матричному виду. Для этого используется множество методов, но в данном случае можно взять метод моментов.

Метод моментов состоит по существу в сведении исследуемого интегрального уравнения к системе линейных алгебраических уравнений с N неизвестными, которые обычно представляют собой коэффициенты некоторого разложения для тока.

Решение задачи можно свести к четырем этапам:

1. Искомый вектор J разлагается в ряд по базисным функциями J_n в области определения оператора $L_{оп}$.

2. Определяется подходящее внутреннее произведение и устанавливается система весовых функций.

3. Вычисляются внутренние произведения и тем самым уравнения приводятся к матричному виду.

4. Находится решение матричного уравнения.

Для решения задач об излучении электромагнитных волн очень удобно использовать вариант метода моментов – метод Галеркина. На первом этапе можно представить функцию отклика J в виде разложения по системе базисных функций $J_1, J_2, J_3, \dots$, заданных на поверхности S и принадлежащих к области определения оператора $L_{оп}$:

$$J = \sum_n I_n J_n, \quad (1)$$

где коэффициенты I_n могут быть комплексными.

Воронцов Глеб Владимирович – ВИВТ АНОО ВО, студент, e-mail:voroncgnrewcbjk19@yandex.ru.

Решение задачи в итоге сводится к определению неизвестных коэффициентов I_n. Если они найдены – значит, известны и амплитуда, и фаза тока на излучателе.

Если ввести оператор

$$L_{\text{оп}}(J) = -(E^S), \qquad (2)$$

и воспользоваться понятием линейных векторных пространств и операторов и записать уравнение в операторном виде:

$$L_{\text{оп}}(J) = (E^i), \qquad (3)$$

где оператор $L_{\text{оп}}$ должен определяться для каждой конкретной задачи, (E^i) – заданная функция возбуждения (источник), J – искомая функция отклика, то подставив разложения (1) и (3), получим, в силу линейности оператора,

$$\sum_n I_n L_{\text{оп}}(J_n) = (E^i). \qquad (4)$$

Второй же этап решения состоит во введении системы весовых функций $W_1, W_2, \ldots$, принадлежащих области определения $L_{\text{оп}}$, и образовании внутреннего произведения

$$\sum_n I_n \langle W_m, L_{\text{оп}} J_n \rangle = \langle W_m, E^i \rangle. \qquad (5)$$

В случае, когда весовые функции совпадают с базисными, $W_m = J_m$, метод называют методом Галеркина, и уравнение (5) принимает следующий вид:

$$\sum_n I_n \langle J_m, L_{\text{оп}} J_n \rangle = \langle J_m, E^i \rangle. \qquad (6)$$

На третьем этапе решения вычисляются различные внутренние произведения, входящие в уравнение (6). Это приводит к матричному уравнению (СЛАУ в матричном виде)

$$\begin{bmatrix} \langle J_1, L_{\text{оп}} J_1 \rangle & \langle J_1, L_{\text{оп}} J_2 \rangle & \ldots \\ \langle J_2, L_{\text{оп}} J_1 \rangle & \ldots & \\ \ldots & & \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ \ldots \\ I_n \end{bmatrix} = \begin{bmatrix} \langle J_1, E^i \rangle \\ \langle J_2, E^i \rangle \\ \ldots \\ \langle J_n, E^i \rangle \end{bmatrix}, \qquad (7)$$

которое можно записать в более компактном виде:

$$[Z](I) = (V). \qquad (8)$$

Элементы матрицы $[Z]$ имеют смысл обобщенных импедансов (сопротивлений), а элементы столбцов (I) и (V) – обобщенных токов и напряжений.

Решение матричных уравнений на ЭВМ обычно производится обращением матрицы или методом исключения ведущего элемента, или методом итераций. Причем, больше всего используется для решения матричных уравнений на ЭВМ, именно метод исключения ведущего элемента.

Для решения СЛАУ (7) можно воспользоваться несколькими методами, которые, в свою очередь, можно распараллелить для вычислений на ЭВМ. Самым оптимальным методом в данном случае является LU-разложение: для произвольной матрицы A существует разложение

$$A = LU, \qquad (9)$$

где L и U – соответственно нижняя и верхняя треугольные матрицы. А с учетом (8) уравнение (9) принимает вид

$$LUI = V. \qquad (10)$$

Если же обозначить $Y = UI$, то решение уравнения 10 находится из последовательного решения двух систем с треугольными матрицами, что хорошо еще сказывается на хранении этих матриц в оперативной памяти [14]. Дело в том, что для хранения в оперативной памяти ЭВМ треугольной матрицы необходимо примерно в два раза меньше памяти ($\frac{m(m+1)}{2}$ ячеек памяти, где m – длина массива), чем для хранения квадратной. Сначала решается система

$$LY = V, \qquad (11)$$

затем решается система линейных уравнений с верхней треугольной матрицей

$$UI = Y. \qquad (12)$$

После же получения разложения (9) для матрицы A, можно использовать его для решения системы уравнений с данной матрицей и разными значениями правой части B. Это осуществляется итерационно путем последовательного изменения правой части уравнения (7), по результатам решения системы на каждой итерации. А в случае использования метода Гаусса, еще одного метода для решения СЛАУ, необходимо на каждой итерации заново выполнять прямой и обратный ход данного метода. При использовании LU-разложения необходимо только на каждой итерации решать уравнение (10).

Можно проанализировать эти два метода решения СЛАУ на тему оценки количества арифметических операций, чтобы понять, что более эффективно можно распараллелить. Количество арифметических операций при решении СЛАУ методом Гаусса оценивается как $O(n^3)$ [15], где n – количество переменных. Для ленточной матрицы с полушириной ленты k оценка принимает значение $O(nk^2)$.

Оценка количества арифметических операций для ленточной матрицы, необходимых для реализации LU-разложения, составит $O(nk^2)$, а для решения на его основе системы (7) – $O(nk)$ [15].

Как следует из приведенных оценок выше, вычислительная сложность метода Гаусса и метода LU-разложения практически одинакова. Однако, если необходимо решить несколько систем с одинаковыми матрицами коэффициентов, но различными векторами свободных членов, то метод LU-разложения окажется более оптимальным, так как в этом случае нет необходимости производить разложение матрицы коэффициентов многократно. Достаточно лишь сохранить полученные треугольные матрицы в памяти (а они занимают не так много места, как было сказано выше) и, подставляя различные вектора свободных членов, получать решения методами прямой и обратной подстановки.

Возьмем для алгоритма параллельного решения матричного уравнения метод LU-разложения, причем алгоритм предполагает, что система линейных уравнений решается без выбора ведущего элемента.

Для решения системы линейных уравнений (7) матрица A распределяется на p процессоров (потоки) [16, 17] следующим образом

$$\begin{pmatrix} A_1 & B_1^U & & & & \\ B_1^L & C_1 & D_2^U & & & \\ & D_2^L & A_2 & B_2^U & & \\ & & \ddots & \ddots & \ddots & \\ & & & B_{p-1}^L & C_{p-1} & D_p^U \\ & & & & D_p^L & A_p \end{pmatrix} \begin{pmatrix} x_1 \\ \xi_1 \\ x_2 \\ \vdots \\ \xi_{p-1} \\ x_p \end{pmatrix} = \begin{pmatrix} b_1 \\ \beta_1 \\ b_2 \\ \vdots \\ \beta_{p-1} \\ b_p \end{pmatrix}, \qquad (13)$$

где A_i – матрица размерности $n_i \times n_i, i = 1, \dots, p$; C_i – матрица размерности $k \times k$; x_i, b_i – вектора размерности n_i; ξ_i, β_i – векторы размерности k; $\sum_{i=1}^{p} n_i + (p-1)k = n, k$ – полуширина ленты матрицы. Такое трехдиагональное блочное разбиение возможно только в случае, если $\forall i n_i > k$. Это условие ограничивает предельно возможный уровень параллелизма: максимальное количество процессоров (потоков) p, которое может быть использовано для параллельного исполнения, должно удовлетворять условию $p < \frac{n+k}{2k}$.

Далее над матрицей производится первый шаг блочно-циклической редукции, при котором переставляются местами строки и столбцы таким образом, чтобы сначала шли все нечетные строки и столбцы, а потом все четные. В итоге, получаем так называемую четно-нечетную перестановку

$$\begin{pmatrix} A_1 & & & & & B_1^U & & & \\ & A_2 & & & & D_2^L & B_2^U & & \\ & & \ddots & & & & \ddots & \ddots & \\ & & & A_{p-1} & & & & \ddots & B_p^U \\ & & & & A_p & & & & D_p^L \\ B_1^L & D_2^U & & & & C_1 & & & \\ & B_2^L & \ddots & & & & C_2 & & \\ & & \ddots & \ddots & & & & \ddots & \\ & & & B_{p-1}^L & D_p^U & & & & C_{p-1} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{p-1} \\ x_p \\ \xi_1 \\ \xi_2 \\ \vdots \\ \xi_{p-1} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{p-1} \\ b_p \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{p-1} \end{pmatrix}. \qquad (14)$$

Равенство (14) можно записать в более общей форме

$$\begin{pmatrix} \mathrm{A} & B^U \\ B^L & C \end{pmatrix}\begin{pmatrix} x \\ \xi \end{pmatrix} = \begin{pmatrix} b \\ \beta \end{pmatrix}. \quad (15)$$

LU-разложение матрицы A имеет вид:

$$\begin{pmatrix} \mathrm{A} & B^U \\ B^L & C \end{pmatrix} = \begin{pmatrix} L & 0 \\ B^L U^{-1} & 1 \end{pmatrix}\begin{pmatrix} U & L^{-1}B^U \\ 0 & S \end{pmatrix}, \quad (16)$$

где $S = C - B^L \mathrm{A}^{-1} B^U$ – дополнение Шура [18] для матрицы A.

Для экономии памяти матрицы $B_i^L U_i^{-1}, L_i^{-1} B_i^U, D_i^U R_i^{-1}$ и $L_i^{-1} D_i^L$ помещаются в те же ячейки памяти, которые выделены для матриц $B_i^L, B_i^U, D_i^U, D_i^L$ соответственно. Так как в процессе вычислений в матрицах D_i^U и D_i^L происходит дополнительное заполнение, то требуется дополнительное место в памяти для $2kn$ чисел с плавающей точкой. В итоге, общее требование к памяти примерно в два раза больше, чем для последовательного алгоритма.

Дополнение Шура S в матрице A для A представляет собой блочную трехдиагональную матрицу размерности $(p-1)\times(p-1)$ с размером блоков $k \times k$:

$$S = \begin{pmatrix} T_1 & U_2 & & & \\ V_2 & T_2 & U_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & U_{p-1} \\ & & & V_{p-1} & T_{p-1} \end{pmatrix}, \quad (17)$$

где

$$T_i = C_i - B_i^L A_i^{-1} B_i^U - D_{i+1}^U A_{i+1}^{-1} D_{i+1}^L = C_i - \left(B_i^L R_i^{-1}\right)\left(L_i^{-1} B_i^U\right) - \left(D_{i+1}^U R_{i+1}^{-1}\right)\left(L_{i+1}^{-1} D_{i+1}^L\right)$$

$$, U_i = -\left(D_i^U R_i^{-1}\right)\left(L_i^{-1} B_i^U\right), V_i = -\left(B_i^L R_i^{-1}\right)\left(L_i^{-1} D_i^L\right).$$

Используя разложение (16), из уравнения (15) получаем:

$$\begin{pmatrix} R & L^{-1}B^U \\ & S \end{pmatrix}\begin{pmatrix} x \\ \xi \end{pmatrix} = \begin{pmatrix} L & \\ B^L R^{-1} & I \end{pmatrix}^{-1}\begin{pmatrix} b \\ \beta \end{pmatrix} = \begin{pmatrix} L^{-1} & \\ -B^L R^{-1} L^{-1} & I \end{pmatrix} = \begin{pmatrix} c \\ \gamma \end{pmatrix}, \quad (18)$$

где части c_i и β_i векторов c и β задаются, как

$$c_i = L_i^{-1} b_i$$

$$\gamma_i = \beta_i - B_i^L R_i^{-1} c_i - D_{i+1}^U R_{i+1}^{-1} c_{i+1}$$

До этого момента не требуется межпроцессорной (или межпоточной, если используется множество потоков, вместо множества процессоров) коммуникации, так как каждый процессор (поток) независимо вычисляет разложение блока диагонали $A_i = L_i R_i$ и вычисляет блоки $B_i^L U_i^{-1}, L_i^{-1} B_i^U, D_i^U R_i^{-1}, L_i^{-1} D_i^L$ и $L_i^{-1} b_i$. Каждый процессор или поток формирует свою часть редуцированной системы $S\xi = \gamma$, где

$$S_i = \begin{pmatrix} -D_i^U R_i^{-1} L_i^{-1} D_i^L & -D_i^U R_i^{-1} L_i^{-1} B_i^U \\ -B_i^L R_i^{-1} L_i^{-1} D_i^L & C_i - B_i^L R_i^{-1} L_i^{-1} B_i^U \end{pmatrix}$$

$$\gamma_i = \begin{pmatrix} -D_i^U R_i^{-1} c_i \\ \beta_i - B_i^L R_i^{-1} c_i \end{pmatrix}.$$

Матрица S_i имеет размерность $2k \times 2k$, векторы ξ_i и γ_i имеют размерность $2k$.

После выполнения локальных вычислений для завершения построения матрицы T_i процессор (поток) с номером i посылает процессору (потоку) с номером $i+1$ матрицу $C_i - B_i^L R_i^{-1} L_i^{-1} B_i^U$.

Четно-нечетная перестановка, вычисление разложения и формирование редуцированной системы повторяется до тех пор, пока матрица редуцированной системы не станет заполненной матрицей размерностью $k \times k$. В этом случае, получившиеся система может быть решена на одном процессоре или потоке. Так как на каждой итерации редукции размерность матрицы уменьшается в два раза, то для достижения результата необходимо выполнить $\log_2 p - 1$ шагов.

После нахождения векторов ξ_i, процессоры (потоки) вычисляют свою часть вектора x:

$$x_1 = R_1^{-1}(c_1 - L_1^{-1} B_1^U \xi_1)$$
$$x_i = R_i^{-1}\left(c_i - L_i^{-1} D_i^1 \xi_{i-1} - L_i^{-1} B_i^U \xi_i\right)$$
$$x_p = R_p^{-1}\left(c_p - L_{p-1}^{-1} D_p^L \xi_{p-1}\right)$$

После нахождения численного решения, остается только найти остальные параметры: распределение тока в антенне, входной импеданс, диаграмму направленности и т. д., что уже не требует такого количества времени и вычислительных ресурсов.

ЛИТЕРАТУРА

1. Глотова Т. В. Применение методов оптимизации для проектирования поглотителей электромагнитных волн / Т. В. Глотова // Моделирование, оптимизация и информационные технологии. – 2016. – № 1. – С. 8.

2. Глотова Т. В. Применение гибридного метода для расчета характеристик рассеяния объектов над шероховатой поверхностью / Т. В. Глотова // Моделирование, оптимизация и информационные технологии. – 2016. – № 1. – С. 11.

3. Преображенский А. П. Моделирование характеристик рассеяния объектов, в состав которых входят кромки / А. П. Преображенский // Моделирование, оптимизация и информационные технологии. – 2016. – № 2. – С. 7.

4. Юрочкин А. Г. Анализ приближенной модели для оценки средних характеристик рассеяния дифракционной структуры / А. Г. Юрочкин, А. В. Данилова, И. А. Гусарова // Моделирование, оптимизация и информационные технологии. – 2016. – № 2. – С. 10.

5. Щербатых С. С. Метод интегральных уравнений как основной способ анализа в САПР антенн / С. С. Щербатых // Моделирование, оптимизация и информационные технологии. – 2016. – № 1. – С. 10.

6. Глотова Т. В. Модификация метода моментов в задачах рассеяния электромагнитных волн / Т. В. Глотова, Т. В. Мельникова // Моделирование, оптимизация и информационные технологии. – 2016. – № 2. – С. 11.

7. Казаков Е. Н. Разработка и программная реализации алгоритма оценки уровня сигнала в сети WI-FI / Е. Н. Казаков // Моделирование, оптимизация и информационные технологии. – 2016. – № 1. – С. 13.

8. Бокова О. И. Проектирование наземных радиосистем передачи информации с помощью специализированных программных комплексов / О. И. Бокова, С. В. Канавин, Н. С. Хохлов // Моделирование, оптимизация и информационные технологии. – 2016. – № 2. – С. 6.

9. Бокова О. И. Повышение быстродействия устройств аналогоцифрового приема и обработки сигналов широкополосных комплексов пеленгования / О. И. Бокова, Д. А. Жайворонок, О. С. Сластникова // Моделирование, оптимизация и информационные технологии. – 2016. – № 2. – С. 9.

10. Юрочкин А. Г. Возможности использования итерационного метода при расчетах характеристик рассеяния комбинированных объектов / А. Г. Юрочкин, А. В. Данилова, И. А. Гусарова // Моделирование, оптимизация и информационные технологии. – 2016. – № 2. – С. 13.

11. Мохаммад М. И. О реализации концепции рекурсивно-параллельного программирования / М. И. Мохаммад, А. В. Данилова // Моделирование, оптимизация и информационные технологии. – 2016. – № 2. – С. 16.

12. Lvovich I. Ya. The analysis of scattering electromagnetic waves with use of paraller computing / I. Ya. Lvovich, A. P. Preobrazhenskiy, O. N. Choporov, K. V. Kaydakova // В сборнике: 2015 International Siberian Conference on Control and Communications, SIBCON 2015 – Proceedings 2015. – С. 714-733.

13. Преображенский А. П. О возможностях ускорения вычислений при решении задач / А. П. Преображенский // Вестник Воронежского института высоких технологий. – 2014. – № 12. – С. 67-68.

15. Бахвалов Н. С. Численные методы. / Н. С. Бахвалов // М., Наука, 1975. – 632 с.

14. Амосов А. А. Вычислительные методы для инженеров: Учеб. пособие / А. А. Амосов, Ю. А. Дубинский, Н. В. Копченова // М.: Высш. шк., 1994. – 544 с.

16. Cleary A., Dongarra J. Implementation in ScaLAPACK of Divide-and-Conquer Algorithms for Banded and Tridiagonal Linear Systems / A. Cleary, J. Dongarra // Computer Science Dept. Technical Report S. 97-358, 1997.

17. Arbenz P. Comparison of parallel solvers for diagonally dominant and general narrow-banded linear systems / P. Arbenz, A. Cleary, J. Dongarra, M. A. Hegland // Tech. Report 312, ETH Zurich, Computer Science Department, 1999

18. Кормен Т. Х. Алгоритмы: построение и анализ. / Т. Х. Кормен, Ч. И. Лейзерсон, Р. Л. Ривест, К. Штайн. – М., Вильямс, 2008. – 1296 с.

THE APPLICATION OF METHODS FOR DISTRIBUTED CALCULATIONS WHEN DESIGNING ELECTRODYNAMIC OBJECTS

© 2016 G. V. Vorontsov

This paper describes the estimation algorithm of scattering characteristics of electromagnetic objects based on the method of integral equations. It is stated that to speed up calculations can be used mechanisms of parallelization.

Keywords: communications, electrodynamics object, parallel computing.