

ВОЗМОЖНОСТИ УСКОРЕНИЯ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ЭФФЕКТА ВТОРИЧНОЙ МОДУЛЯЦИИ СИГНАЛА БОРТОВОЙ РАДИОЛОКАЦИОННОЙ СТАНЦИИ В СРЕДЕ MATLAB

А. А. Потапов¹, В. А. Кузнецов², Н. А. Меньших³

¹ Институт радиотехники и электроники им. В.А. Котельникова РАН, Москва

² Военно-воздушная академия им. проф. Н.Е. Жуковского и Ю. А. Гагарина, Воронеж

³ Военный инновационный технополис «ЭРА», Анапа

В работе кратко рассмотрена возможность оптимизации кода в среде MATLAB. Предложено два подхода для ускорения работы кода MATLAB. Применение предложенного способа позволяет ускорить вычисления при задействовании малого числа ядер процессора.

Ключевые слова: радиолокационная станция, имитационное моделирование, параллельные вычисления, вторичная доплеровская модуляция.

Среди современных прикладных пакетов математического моделирования наибольшее распространение для реализации исследовательских задач получил MATLAB, представляющий широкие возможности численного анализа. Для того чтобы обеспечить совместимость с существующими проектами, MATLAB обеспечивает возможность переводить код на другие языки программирования, например, Fortran или C [1]. Большое количество библиотек MATLAB позволяют моделировать явления самой разной природы, однако крупномасштабные численные расчеты в среде MATLAB предъявляют к аппаратным вычислительным средствам высокие требования. Пользователь сталкивается с одним из двух вариантов решения данной проблемы: либо произвести модернизацию аппаратных средств, либо оптимизировать алгоритмы вычислений. Первый способ имеет множество различных недостатков: во-первых, пользователь не всегда может быстро освоить вычислительную мощность в полной мере, во-вторых, имеет временный характер так как каждый год появляются новые технологии для увеличения производительности компьютера. Поскольку все потенциальные возможности MATLAB реализуются благодаря обширному набору библиотек и операций векторизации, существует другой способ для ускорения вычисления – это оптимизация кода для эффективного выполнения алгоритма.

Задача имитационного моделирования радиолокационной воздушной цели сама по себе является достаточно сложной в вычислительном плане, а моделирование эффекта вторичной модуляции сигнала бортовой радиолокационной станции, отраженного от вращающихся лопаток компрессора низкого давления турбореактивного двигателя воздушной цели в процессе ее движения – в особенности. В работе используется многолучевой подход к описанию распространения электромагнитной энергии с учетом ее многократного переотражения от элементов фюзеляжа, что занимает весьма значительное время и обуславливает целесообразность ускорения имитационного моделирования.

Целью данной работы является оценка времени имитационного моделирования эффекта вторичной модуляции радиолокационного сигнала при использовании центра обработки данных (ЦОД), а также требуемого количества вычислительных ресурсов.

Для ускорения работы кода MATLAB известно два подхода: предварительное размещение (preallocation) и векторизация (vectorization). С использованием предварительного размещения осуществляется инициализация массива окончательного размера, требуемого для вычислений. Предварительное размещение помогает избегать динамического изменения размера массивов, особенно если код содержит циклы for и while. Поскольку массивы в MATLAB находятся в непрерывных блоках памяти, повторное изменение размера массива требует от MATLAB дополнительных затрат времени на поиск большего блока памяти и последующего перемещения массива в этот блок памяти. Использование предварительного размещения массива позволяет избежать этих ненужных операций и сократить общее время выполнения вычислений.

Векторизация – это процесс перехода от использования циклов к матричным и векторным операциям. MATLAB позволяет пользователю использовать процессорно-оптимизированные библиотеки матричных и векторных вычислений. В результате можно увеличить производительность посредством векторизации кода [2]. Векторизованные вычисления MATLAB, использующие большие массивы, позволят пользователю ускорить вычисления с использованием графического процессора (GPU – graphics processing unit). В случаях, когда циклы for не могут быть векторизованы, для ускорения алгоритма используются параллельные циклы for (parfor) или генерация кода на языке программирования C.

В связи с развитием многопроцессорности и многоядерности, параллельные вычисления превратились из экзотики в повседневность, однако многие вычислительные среды, в частности, MATLAB, не поддерживают многопоточность по умолчанию и требуют дополнительных инструкций и модулей. Многоядерные параллельные вычисления производятся на основе MPI (Message Passing Interface) интерфейса, который реализует упрощенный переноса программы с кластера на кластер, не изменяя ни байта исходного кода. Параллельная программа должна эффективно использовать вычислительные мощности и коммуникационную среду.

Генерация кода на языке C предоставляет несколько иной подход к ускорению – вместо распределения вычислительной задачи по множеству ядер процессора производится анализ этой задачи и выделение наиболее часто используемых в ней функций. Функции языка MATLAB имеют формат М-функций. М-функции являются М-файлами, которые допускают наличие входных и выходных аргументов. Они работают с переменными в пределах собственной рабочей области, отличной от рабочей области системы MATLAB. Встроенные М-функции являются скомпилированными и выполняются достаточно быстро, однако, существует возможность создания MEX-функций, реализованных на языке C/C++, из существующих М-функций с помощью специального интерфейса, называемого MATLAB Coder. MEX-функции ведут себя подобно встроенным функциям MATLAB, но их разработка требует точно определять типы и размеры аргументов, а также учитывать различия между языками MATLAB и C.

Рассмотрим псевдо-алгоритм для расчета вторичной доплеровской модуляции (ВДМ) сложных воздушных целей, который состоит из следующих частей:

1. Загрузка модели воздушной цели и определение характеристик процесса радиолокации.
2. Расчет параметров для расчета ВДМ – таких как длина волны сигнала РЛС, количество лучей, направляемых от РЛС в сторону воздушной цели, мощность одного луча и т. д.
3. Установка дополнительных параметров расчета: параметров канала, а также алгоритма расчета ВДМ с учетом эффективной поверхности рассеяния (ЭПР) (например, плотность водяного пара в пределах канала и атмосферное давление).
4. Расчет ВДМ с учетом ЭПР.

Расчет ВДМ в указанном алгоритме проводится на основе 3D-модели, представленной в формате STL (stereolithography – стереолитография, так как изначально этот формат применялся в области трехмерной печати). Данный формат удобен тем, что его можно использовать для 3D-моделей во многих популярных системах автоматизированного проектирования (Solidworks, Autodesk Inventor, Blender). Формат STL описывает объект с помощью списка треугольных граней (фацетов) и нормалей к ним. Для каждого фацета и нормали указываются три числа с плавающей запятой, представляющие координаты в декартовой системе координат. При этом нормаль фацета должна быть единичным вектором, направленным от фацета наружу.

Отметим, что пункты 2 и 3 выполняются единожды для каждой трехмерной модели, а пункт 4 является составным и несет основную нагрузку на вычислительное устройство как в плане потребляемой памяти, так и в плане числа операций вычислений. Очевидно, что именно эта часть алгоритма заслуживает наибольшего внимания.

Оценка времени расчета вторичной доплеровской модуляции подразумевает зависимость этого времени от некоторых исходных данных. Рассматривая сложность и детализацию объекта, выраженную в количестве facets, оценим данную зависимость по нескольким экспериментам. Для этого проведена серия расчетов для STL-моделей различной сложности (с различным количеством facets) на различных процессорах с использованием последовательных и параллельных вычислений. В экспериментах использовались два типа процессора – Intel® Core™ i7-6850K CPU @ 3.60GHz, установленный на штатном рабочем месте, а также Intel® Xeon® Gold 6150 CPU @ 2.70GHz, установленный на серверах центра обработки данных (ЦОД) Военного инновационного технополиса «ЭРА» [3]. Второй процессор имеет 18 ядер [4], что предоставляет возможность еще большего распараллеливания вычислений. Общее время расчета в каждом эксперименте разделено на суммарное количество расчетов ЭПР отдельного facets и тем самым получено среднее время расчета для одного facets. Результаты расчетов представлены в табл. 1 и на рис. 1.

Таблица 1

Время вычисления в секундах для различного количества facets

№	Тип процессора и количество используемых при расчетах ядер	Число facets		
		35590	51364	64703
1	Intel® Core™ i7-6850K CPU @ 3.60GHz – 1 (APM)	41	72	76
2	Intel® Core™ i7-6850K CPU @ 3.60GHz – 6 (APM)	13	24	30
3	Intel® Xeon® Gold 6150 CPU @ 2.70GHz – 1 (ЦОД)	10	45	57
4	Intel® Xeon® Gold 6150 CPU @ 2.70GHz – 6 (ЦОД)	9	17	23

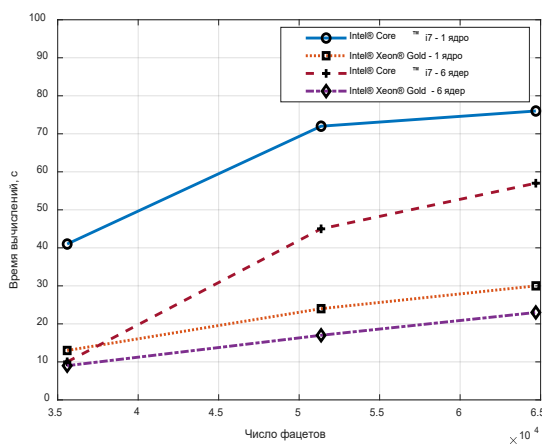


Рис. 1. Зависимость времени вычисления от количества facets модели на различных процессорах и их конфигурациях

Приведенные на рис. 1 графики демонстрируют, что эффективность параллельных вычислений возрастает с количеством facets. Этот результат вполне ожидаем. Вторым выводом является то, что эффективность параллельных вычислений, произведенных на различных процессорах отличается незначительно. Данный результат ставит под сомнение необходимость проведения расчетов с привлечением ЦОД.

Поскольку расчет ВДМ не выражается в виде четкой зависимости от количества facets – скрывается различие в количестве видимых facets с определенного ракурса и числа переотражений лучей – рассмотрим обобщенную структуру алгоритма с абстрактными «элементарными» в данном контексте операциями.

Рассмотрим обобщенную внутреннюю организацию расчета ВДМ:

```

for i=1:n0 %цикл по списку азимутальных углов
%формирование сетки лучей
for j=1:n1 %цикл по числу отсчетов на интервале наблюдения
%выделение характеристик facets
for k=1:n2 % цикл по числу излученных лучей
while A %проверка условий пересечения лучей и facets
%оценка пересечений
%оценка переотражений лучей – учет потерь энергии
if B %хотя бы одно пересечение имеется
for z=1:n4 % цикл по пересеченным элементам модели
% расчет ЭПР видимого facets с учетом поляризации
end
end
end
end
end
% сбор отраженных лучей
end
% вычисление спектра ВДМ
% построение спектрограммы
end
end

```

Проанализируем приведенную структуру для определения подходящего варианта ускорения имитационного моделирования эффекта вторичной модуляции:

1. Расчет ЭПР видимого facets с учетом поляризации является абстрактной единицей приведенного алгоритма. Расчет ЭПР производится в М-функции, следовательно, возможно ускорение алгоритма путем преобразования рассматриваемой m-функции в МEX-функцию [5].

2. Цикл по числу излученных лучей содержит независимые итерации, а прохождение каждого луча оценивается независимо от других. Это позволяет говорить о возможности использования вместо обычного цикла for цикла parfor [6].

Воспользуемся приведенными особенностями алгоритма и преобразуем его с использованием МEX-функции и цикла parfor. Чтобы проанализировать эффективности каждого из изменений, проведем серию расчетов как с использованием МEX-функции, так и с использованием ее аналога m-функции, при этом варьируя количество используемых ядер в цикле parfor. Для упрощения расчеты будем проводить для одного азимутального угла на 127 отсчетах на интервале наблюдения и количестве излученных лучей, равном 289. Для сравнения эффективности совместного применения parfor и МEX-функций, проведем серию расчетов (по 10 на каждую модификацию) и сравним полученные результаты (табл. 2).

Таблица 2

Результаты экспериментальных оценок времени расчетов ВДМ при различных модификациях параметров алгоритма расчета

№	Использование parfor (количество используемых ядер процессора)	Использование МEX-функции	Усредненное время расчета, с
1	– (1)	–	3544.215506
2	+ (2)	–	2300.645179
3	+ (3)	–	1712.494566
4	+ (4)	–	1480.251042
5	+ (5)	–	1444.556580
6	+ (6)	–	1380.726564
7	– (1)	+	2221.899430
8	+ (2)	+	1508.524474
9	+ (3)	+	1225.291389
10	+ (4)	+	1114.454955
11	+ (5)	+	1096.561757
12	+ (6)	+	1052.225927

При параллелизации алгоритма вычисления проводились на ядрах процессора Intel Core i5-8400 CPU 2,800 ГГц. Количество используемых ядер варьировалось для анализа влияния МЕХ-функции на изменение времени расчета. Для наглядности и удобства анализа представим табличные данные в виде графика (рис. 2).

Легко заметить, что полученные зависимости имеют экспоненциальный характер, причем использование более 4 ядер в параллельных вычислениях уже не имеет такого принципиального значения в плане повышения скорости выполнения кода. Для оценки максимально достижимого параллелизма уместно воспользоваться законом Амдала, который описывает связь между ожидаемым ускорением параллельных реализаций алгоритма и последовательным алгоритмом в предположении, что решаемая задача – в данном случае исходные данные для моделирования, остается постоянной [7]. Допущение о фиксированном размере задачи является ключевым для закона Амдала – если количество обрабатываемых данных меняется с варьированием количества ядер, следует обратиться к другой зависимости, например, закону Густафсона [8].

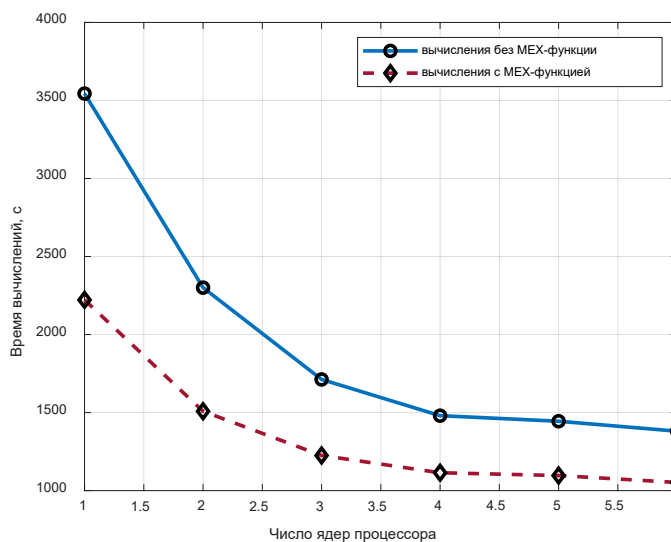


Рис. 2. Зависимость времени выполнения расчетов от количества используемых ядер

Введем некоторые необходимые для анализа величины. Ускорением параллельного алгоритма называется отношение:

$$S_n = \frac{T_1}{T_n} \quad (3)$$

где T_n – время вычисления задачи на n процессорах, T_1 – время выполнения однопоточной программы. Если $S_n > 1$, то параллельная версия алгоритма эффективна, в противном случае накладные расходы (издержки) реализации параллельной версии алгоритма чрезмерно велики. Закон Амдала описывается следующей математической формулой:

$$S_n = \frac{1}{\alpha + \frac{1-\alpha}{n}}, \quad (4)$$

где S_n показывает, во сколько раз можно ускорить вычисления (ускорение), n – количество процессоров (ядер), α – доля последовательно вычисляемого кода.

Закон Амдала, несмотря на то, что в нем не учитывается множество факторов, накладывает ограничения на максимально достижимую эффективность параллельного алгоритма. Оценим долю последова-

тельной части программы по закону Амдала и спрогнозируем выигрыш в производительности при использовании большего числа процессоров. Проведем две оценки доли последовательной части программы: с использованием MEX-функции (α_2) и без таковой (α_1). Рассчитаем для этого указанные доли для каждой параллельной конфигурации из таблицы 1 и выведем среднее значение для α_2 и α_1 (табл. 3).

Таблица 3

Расчетные значения последовательной доли программы расчета ВДМ

Кол-во используемых ядер	α_1	α_2
2	0.2983	0.3579
3	0.2248	0.3272
4	0.2235	0.3354
5	0.2595	0.3669
6	0.2675	0.3683
Среднее значение α	0.2547	0.3511

Полученные значения характеризуют эффект от применения MEX-функций – доля последовательно вычисляемого кода в среднем увеличивается в 1,3785 раз. Начиная с некоторого числа процессов, зависящего от доли последовательной части, ускорение будет меняться незначительно и дальнейшее распараллеливание не имеет смысла. Максимальное достижимое ускорение не может превысить в случае применения обычных m-функций $S_{\max1} = 1/\alpha_1 = 3,9261$, а в случае применения MEX-функции – $S_{\max2} = 1/\alpha_2 = 2,8482$. Заметим, что в первом случае при использовании 6 ядер достигнуто ускорение в 2,5669 раз – 65 % от теоретического максимума, а во втором случае при использовании 6 ядер достигнуто ускорение в 2,1116 – 74 % от теоретического максимума. Если рассматривать возможность расчета на процессоре ЦОД с использованием всех имеющихся 18 ядер, то получим значения ускорения $S_1 = 3.3772$ (86 % от теоретического максимума) и $S_2 = 2.5830$ (90 % от максимума). Оценивая полученные показатели эффективности, выраженные в виде доли полученного ускорения от теоретически возможного, получим вывод, что трехкратное увеличение количества ядер дает прирост ускорения всего на 16 %, поэтому выполнение параллельных вычислений на шести процессорах можно считать приемлемым и достаточным.

Таким образом, к программе расчета спектра ВДМ была применена стратегия ускорения вычислений в среде MATLAB, основной упор в которой ставился на совместном применении параллельных циклов parfor и MEX-функций. Результаты проведенных экспериментов показали, что значительная доля возможного ускорения достигается при умеренном количестве используемых ядер процессора, следовательно, применение процессора ЦОД не имеет ключевого значения. В дальнейшем планируется анализ эффективности применяемых структур данных и перевод вычислений с процессора на GPU. Подобные преобразования будут призваны открыть новые возможности для векторизации и распараллеливания вычислительных задач в среде MATLAB.

Литература

1. Lopez, C. MATLAB Numerical Calculations / C. Lopez.–New York: Apress, 2014.–336 p.
2. Chapra, S. Applied Numerical Methods with MATLAB® for Engineers and Scientists: Fourth Edition / S. Chapra.–New York: McGraw-Hill Education, 2017.–720 p.
3. ЭРА вычислений на новом уровне: СуперЭВМ расширила исследовательские возможности технополиса Минобороны [электронный ресурс] Режим доступа: <https://www.era-tehnopolis.ru/news/mass-media/era-vychisleniy-na-novom-urovne/> (дата обращения: 02.01.21.)
4. Процессор Intel® Xeon® 6150 класса Gold [электронный ресурс] Режим доступа: <https://ark.intel.com/content/www/ru/ru/ark/products/120490/intel-xeon-gold-6150-processor-24-75m-cache-2-70-ghz.html> (дата обращения: 02.01.21.)
5. MATLAB Coder Documentation: Code Generation [электронный ресурс] Режим доступа: <https://www.mathworks.com/help/coder/code-generation.html> (дата обращения: 02.01.21.)

6. Parallel Computing Toolbox Documentation: Parallel for-Loops (parfor) [электронный ресурс] Режим доступа: https://www.mathworks.com/help/parallel-computing/parallel-for-loops-parfor.html?s_tid=CRUX_lftnav (дата обращения: 02.01.21.)

7. Белова, И. М. Параллельное программирование: учебное пособие / Белова И. М., Рассказов А. А.–М.: МГИУ, 2012. –101 с.

8. Sun, X.-H. Reevaluating Amdahl's law in the multicore era / X.-H. Sun, Y. Chen // J. J. Parallel and Distributed Computing.–2010.–№ 70.–pp. 183–188.

CAPABILITIES OF ACCELERATING THE SIMULATION OF THE SECONDARY SIGNAL MODULATION EFFECT OF ON-BOARD RADAR IN MATLAB ENVIRONMENT

A. A. Potapov¹, V. A. Kuznetsov², N. A. Menshikh³

¹Kotelnikov Institute of Radioengineering and Electronics RAS, Moscow

²Zhukovsky – Gagarin Air Force Academy, Voronezh

³Military Innovative Technopolis “ERA”, Anapa

The paper briefly describes code optimization capabilities in MATLAB environment. Two approaches are proposed to speed up the MATLAB code operation. The use of the proposed method allows accelerating computations with a small number of enabled processor cores.

Key words: radar, simulation, parallel computations, secondary Doppler modulation.