

УДК 519.612.2

С.П. Куксенко, Р.С. Суровцев

Сравнение вычислительных и аналитических оценок ускорения многократного решения СЛАУ блочным LU-разложением

Представлены результаты оценки производительности некоммерческих библиотек линейной алгебры BOOST и Eigen на примере LU-разложения. Выявлено, что в зависимости от типа переменной и конфигурации проекта Microsoft Visual C++ алгоритм библиотеки Eigen в 3–53 раза быстрее. Выполнена программная реализация алгоритма блочного LU-разложения для многократного решения СЛАУ с использованием функций библиотеки Eigen. Полученные вычислительные оценки сравнены с аналитическими оценками, полученными ранее. Показано, что оценки ускорения схожи по характеру зависимости от количества решаемых СЛАУ, но различны по значениям и характеру зависимости от количества правых частей.

Ключевые слова: многократное решение, СЛАУ, BOOST, Eigen, блочное LU-разложение, аналитические и вычислительные оценки.

doi: 10.21293/1818-0442-2016-19-2-71-75

Часто натурные испытания радиоэлектронной аппаратуры (РЭА) являются довольно дорогостоящими, а их проведение требует существенных временных затрат. Кроме того, по результатам испытаний необходим сложный поиск ошибок проектирования, поскольку зачастую РЭА не проходит испытания с первого раза. Проведение повторных испытаний ведет к росту затрат на проектирование. Решение данной проблемы – предварительное имитационное моделирование, которое в настоящее время получило широкое распространение в различных областях практической и научной деятельности.

Процесс построения модели предполагает следующие этапы [1]: постановка задачи (определение целей расчета, класса решаемых задач, необходимого объема входной и выходной информации, а также допустимых пределов погрешностей результатов); аналитическая обработка (формулировка уравнений, начальных и граничных условий, выбор метода решения, преобразование полученных уравнений к подходящему для данного численного метода виду); дискретизация границ модели (переход от непрерывных функций к дискретным, а также от функциональных уравнений к алгебраическим); решение полученной системы уравнений с заданной точностью; обработка результатов (расчет поля, характеристик и параметров системы по результатам решения и их визуализация). Таким образом, процесс моделирования в конечном счете сводится к решению системы линейных алгебраических уравнений (СЛАУ). Поэтому от умения быстро и с заданной точностью ее решать зависит эффективность моделирования в целом.

Показательным примером моделирования является вычисление методом моментов матрицы погонных коэффициентов электростатической индукции (далее – емкостной матрицы) в диапазоне параметров [2], реализованное в системе TALGAT, разрабатываемой с использованием объектно-ориентированного языка программирования C++ [3]. Метод моментов по существу сводит задачу моделирования к решению СЛАУ вида

$$\mathbf{S}_k \boldsymbol{\Sigma}_k = \mathbf{V}_k, \quad (1)$$

где \mathbf{S}_k – квадратная и плотная матрица порядка N , являющаяся результатом применения метода моментов к анализируемой структуре; \mathbf{V}_k – неизменная в ходе вычислений матрица размера $N \times N_{\text{COND}}$, состоящая из задаваемых потенциалов на подобластях, на которые разбиты границы структуры, а $\boldsymbol{\Sigma}_k$ – искомая матрица размера $N \times N_{\text{COND}}$, дающая распределение плотности заряда на этих границах, N_{COND} – количество правых частей (количество проводников), не считая опорного, $k = 1, 2, \dots, m$ (m – число точек изменения параметров исходной конфигурации). Порядок (N) матрицы СЛАУ складывается из количества подобластей на границах проводник–диэлектрик (N_A) и диэлектрик–диэлектрик (N_D), а элементы матрицы вычисляются из параметров этих подобластей.

Примечательно, что специфика изменения матрицы \mathbf{S}_k от матрицы к матрице такова, что изменение ряда параметров исходной конфигурации проводников и диэлектриков приводит лишь к частичному изменению ее элементов (например, при изменении относительной диэлектрической проницаемости диэлектрика изменяется лишь часть элементов главной диагонали с индексами больше N_A) [4]. Тогда, при моделировании в диапазоне параметров нет необходимости для каждого значения k заново вычислять все элементы матрицы, а достаточно пересчитывать лишь изменяющиеся. Аналогично при вычислении вектора решения на шаге k можно использовать часть результатов, полученных на шаге $k-1$. Это в совокупности является ресурсом для ускорения моделирования в диапазоне параметров. Поэтому для решения СЛАУ при вычислении емкостной матрицы необходим выбор метода, использующего специфику неполного изменения матрицы СЛАУ, поскольку от этого зависит эффективность моделирования в целом.

Большинство традиционных методов решения СЛАУ условно делится на два класса: прямые и итерационные. Прямые приводят к решению за конечное число арифметических операций, при этом их

точность обусловлена машинной точностью (поэтому их также называют точными) [5]. Итерационными называются методы, в которых точное решение может быть получено лишь в результате бесконечного повторения, как правило, однообразных действий. Известен весьма обстоятельный обзор методов решения СЛАУ с анализом их сходимости, ошибок округления, требуемых памяти и количества операций [6]. Для решения СЛАУ с плотной матрицей, как правило, используются прямые методы, такие как метод Гаусса или его модификация, основанная на LU-разложении.

Известно исследование [7], использующее специфику неполного изменения матрицы СЛАУ при решении системы (1). Для ее учета использован один из прямых методов решения СЛАУ – блочное LU-разложение. Главным его отличием от традиционного последовательного LU-разложения является то, что исходная матрица СЛАУ разбивается на четыре подматрицы и в результате разложения матрицы L и U являются блочно-треугольными. Полученные аналитические оценки ускорения показали, что ускорения многократного решения СЛАУ за счет блочного LU-разложения могут достигать 101 и 134 раза при $N = 1000$ и $N_d = 900$ для $N_{COND} = 1$ и 10 соответственно. Недостатком описанных исследований является то, что результаты аналитических оценок не подтверждены оценками, полученными за счет вычислительного эксперимента.

В настоящее время существует ряд специализированных математических библиотек, написанных на C++ и реализующих как эффективные реализации операций с векторами и матрицами, так и готовые функции для решения СЛАУ (например, LU-разложение) [8–12]. В последние годы широкое развитие получает создание и внедрение математических библиотек, эффективно реализующих параллельные алгоритмы векторных и матричных операций. Эта тенденция оправдана, так как широкое применение получили многоядерные процессоры и для достижения высокой производительности необходимо обеспечить загрузку каждого из ядер. Важно отметить, что повышения производительности программного кода можно добиться не только распараллеливанием, но и за счет использования различных режимов компиляции и вычислительных инструментов. Например, бесплатно распространяемая библиотека ATLAS позволяет достичь высокой производительности за счет оптимизации кода на этапе компиляции под конкретную вычислительную станцию [8]. Свободно распространяемая библиотека Eigen широко использует параллелизацию с помощью технологии OpenMP (открытый стандарт для распараллеливания программ), а также поддержку инструкций SSE2 (технологии потоковой обработки целочисленных данных, позволяющей расширить возможности процессора) [9]. Среди примеров подобных библиотек достойна внимания исторически первая некоммерческая библиотека LAPACK [10]. Наконец, необходимо отметить еще одну некоммер-

ческую библиотеку BOOST, которая по существу является собранием целого ряда библиотек, расширяющих язык программирования C++ [11]. BOOST имеет заметную направленность на исследования и расширяемость (метапрограммирование и обобщенное программирование с активным использованием шаблонов). В состав BOOST входит библиотека линейной алгебры uBLAS, к которой в качестве низкоуровневых модулей можно подключить специализированные математические библиотеки, оптимизированные для конкретной вычислительной платформы и позволяющие достичь максимально возможной производительности. Благодаря тщательному подбору, включенные в BOOST библиотеки обладают высокой надёжностью и производительностью.

Примечательно сравнение производительности ряда некоммерческих библиотек, таких как Eigen, ATLAS и LAPACK [13]. Его результаты показали, что библиотека Eigen оптимальна (с точки зрения минимизации временных затрат) при использовании компилятора Microsoft Visual C++ (MSVC), поддерживающего инструкции SSE, и многоядерных рабочих станций. Дополнительные вычисления еще при двух наборах ключей (оптимизация совместно с генерацией кода с помощью инструкций SSE2 и поддержкой OpenMP) показали, что эффективность данной библиотеки в первую очередь повышается за счет возможности использования инструкций SSE2. Однако проведенное сравнение выполнено лишь для одного типа переменных – double, в то время как размер числа в байтах (обусловленный тип переменной) может оказывать влияние на время решения СЛАУ. Кроме того, не выполнено сравнение с BOOST.

Цель данной работы заключается в сравнительной оценке ускорения, полученного аналитически и за счет вычислительного эксперимента, многократного решения СЛАУ за счет блочного LU-разложения для последующего встраивания в TALGAT. Для достижения поставленной цели необходимо решить следующие задачи: выполнить оценку времени LU-разложения алгоритмами из библиотек BOOST и Eigen и выбрать наиболее производительную библиотеку; реализовать алгоритм решения СЛАУ за счет блочного LU-разложения с использованием функций выбранной библиотеки; провести ряд вычислительных оценок ускорения многократного решения СЛАУ и сравнить результаты вычислительных оценок с аналитическими оценками из [6].

Сравнение производительности библиотек BOOST и Eigen на примере LU-разложения

Выполнена оценка времени LU-разложения алгоритмами, реализованными в библиотеках BOOST и Eigen. Для проведения оценок выбран компилятор MSVC, позволяющий использовать OpenMP и инструкции SSE2, а также различные виды оптимизации. Для корректности оценок вычислительные эксперименты выполнены для двух типов данных: double и float. Их основное различие заключается в допустимых значениях и размере в байтах, что может значи-

тельно повлиять на время вычислений. В вычислительном эксперименте использовались 4 различные конфигурации проекта MSVC для выявления наиболее эффективной (табл. 1). Для проведения вычислительного эксперимента использована рабочая станция (PC) с характеристиками: ОС – Microsoft Windows 7 Enterprise, ЦПУ – Intel(R) Core(TM) i7 CPU 970 3.2 GHz, ОЗУ – 24 Гб.

Таблица 1
Используемые конфигурации проекта MSVC
(+ поддержка включена, – поддержка отключена)

Конфигурация	Оптимизация	SSE2	OpenMP
A	+	–	–
B	–	+	–
C	–	–	+
D	+	+	+

Оценено время LU-разложения для порядка матрицы $N = 1000, 2000, 3000, 4000$. Полученные оценки для всех конфигураций проекта сведены в табл. 3. На рис. 1 для примера приведены зависимости времени выполнения LU-разложения от порядка

матрицы СЛАУ для переменных типа float и конфигурации «D».

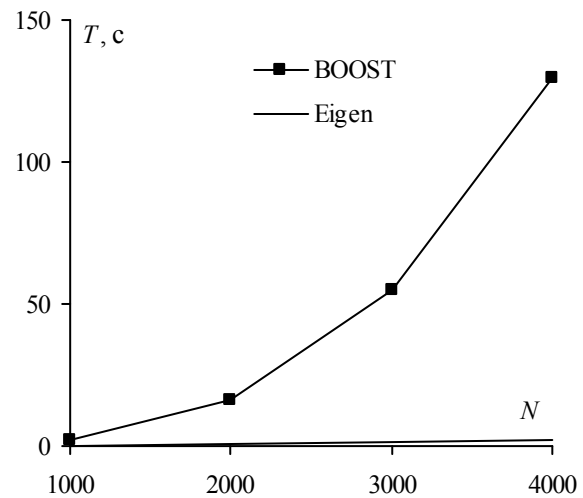


Рис. 1. Зависимости времени LU-разложения от N для переменных типа float и конфигурации «D»

Таблица 3
Время решения (с) СЛАУ при использовании алгоритмов LU-разложения из библиотек BOOST и Eigen

Конфигурация	ПО	N							
		float				double			
		1000	2000	3000	4000	1000	2000	3000	4000
A	BOOST	2,26	18,20	61,56	146,07	2,26	18,44	62,72	148,86
	Eigen	0,68	5,16	17,16	40,25	0,22	1,65	5,41	12,59
B	BOOST	32,09	256,53	865,61	2051,50	32,10	256,87	866,47	2054,33
	Eigen	2,59	20,58	68,55	159,67	4,13	32,03	105,75	248,65
C	BOOST	31,01	247,84	835,53	1981,10	31,53	252,22	850,94	2016,79
	Eigen	3,58	16,77	47,11	95,35	3,47	15,72	42,62	84,35
D	BOOST	2,00	16,14	54,65	129,59	2,25	18,29	62,18	147,64
	Eigen	0,31	0,57	1,17	2,40	0,34	0,96	2,15	4,17

Из табл. 3 видно, что даже для вычислительно мощных PC время LU-разложения матриц больших порядков может быть довольно существенным. Время вычисления с использованием функции библиотеки Eigen существенно меньше, чем BOOST. Так, при использовании только средств оптимизации (конфигурация «A»), время библиотеки BOOST в 3,3–3,6 и 10,3–11,8 раза больше времени библиотеки Eigen для типов переменной float и double соответственно. При поддержке всех инструкций и директив (конфигурация «D») отношение времен библиотек BOOST и Eigen увеличивается до 6,4–53,9 и 6,6–35,4 раза для типов переменной float и double соответственно. Примечательно, что время LU-разложения имеет слабую зависимость от типа используемой переменной только для алгоритма библиотеки BOOST для всех конфигураций, а для алгоритма библиотеки Eigen оказывает неоднозначное влияние. Эта неоднозначность связана со спецификой обработки данных математическим сопроцессором (основной тип переменных – double) и тем, что компиляция оптимального кода для математического сопроцессора является довольно трудной задачей. Наконец, необходимо отметить, что OpenMP и SSE2

вносят несущественный вклад в ускорение вычислений. Наиболее существенное влияние на время LU-разложения оказывает используемая конфигурация проекта. Так, отключение оптимизации кода (конфигурации «B» и «C») приводит к значительному росту затрат времени, который является наиболее существенным для библиотеки BOOST. Таким образом, ключевую роль для минимизации затрат времени на вычисления оказывает оптимизация кода.

Выявленные преимущества библиотеки Eigen говорят о предпочтительности использования ее функций для программной реализации алгоритма блочного LU-разложения по сравнению с функциями библиотеки BOOST.

Оценка ускорения многократного решения СЛАУ блочным LU-разложением

Для оценки эффективности блочного LU-разложения для многократного решения СЛАУ вида (1) при вычислении емкостной матрицы в диапазоне параметров выполнена программная реализация алгоритма блочного LU-разложения с использованием функций библиотеки Eigen. В вычислительном эксперименте использована та же PC. Он выполнен на примере матрицы СЛАУ с $N = 1000$ и $N_D = 900$.

Вычислительные оценки ускорения выполнены при $N_{\text{COND}} = 1, 10$ и $m = 100, 200, \dots, 1000$. В табл. 4 сведены ускорения (отношение времени решения СЛАУ последовательного алгоритма LU-разложения

из библиотеки Eigen и блочного LU-разложения с использованием функций библиотеки Eigen) вычислительного эксперимента, а также аналитические оценки ускорения блочным LU-разложением из [6].

Таблица 4

Аналитические и вычислительные оценки ускорения решения СЛАУ за счет использования алгоритма блочного LU-разложения для вычисления m емкостных матриц

N_{COND}	Оценка	m									
		100	200	300	400	500	600	700	800	900	1000
1	Аналитическая	27,73	49,59	67,26	81,84	94,10	104,52	113,5	121,32	128,18	134,27
	Вычислительная	7,13	14,01	20,09	26,20	30,91	32,33	31,97	36,58	39,21	41,12
10	Аналитическая	26,33	44,68	58,20	68,58	76,80	83,47	88,98	93,63	97,58	101,00
	Вычислительная	7,40	14,47	18,07	24,15	29,96	34,74	37,30	42,89	45,65	46,84

Как видно из табл. 4, результаты аналитических оценок выше (2–4 раза) вычислительных, но характер роста ускорения одинаков: с увеличением m возрастает и ускорение. Необходимо отметить неоднозначное влияние количества правых частей N_{COND} на ускорение решения СЛАУ при аналитических и вычислительных оценках. Примечательно, что при увеличении N_{COND} от 1 до 10 ускорение, полученное аналитически, уменьшается, а при вычислительном эксперименте наблюдается обратная зависимость от N_{COND} – ускорение для всех m при $N_{\text{COND}}=10$ выше, чем при $N_{\text{COND}}=1$. Причины такой зависимости однозначно пояснить сложно. Вероятнее всего, это связано с тем, что при реализации алгоритма блочного LU-разложения предпочтение отдано матричным операциям при нахождении матрицы неизвестных Σ (учтено, что в общем случае количество проводников в системе может быть больше чем 1), в то время как при использовании алгоритма из библиотеки Eigen каждый вектор неизвестных вычисляется отдельно.

Заключение

В работе выполнены оценки производительности двух некоммерческих библиотек линейной алгебры на примере LU-разложения. По результатам проведенных оценок выявлено, что время LU-разложения алгоритмом из библиотеки Eigen в зависимости от типа переменной и конфигурации проекта MSVC может быть в 3–53 раза меньше, чем время обработки алгоритмом из библиотеки BOOST. Поэтому реализация алгоритма блочного LU-разложения для многократного решения СЛАУ выполнена с использованием функции библиотеки Eigen.

Сравнение аналитических и вычислительных оценок ускорения многократного решения СЛАУ за счет блочного LU-разложения показало, что результаты оценок схожи по характеру зависимости от количества точек изменения матрицы СЛАУ m , однако отличаются по характеру зависимости от количества правых частей N_{COND} и по значениям. В дальнейшем необходимо выполнить более детальные оценки ускорения многократного решения СЛАУ блочным LU-разложением. Для этого видится необходимым дополнительно выполнить реализацию алгоритма блочного LU-разложения для многократных вычислений в коммерческом пакете MATLAB и

сравнить полученные ускорения с ускорениями из данной работы. Также целесообразно выполнить сравнение полученных результатов с результатами, полученными с помощью библиотек, разработанных для других языков программирования, например, FORTRAN.

Моделирование методом моментов выполнено за счет гранта Российского научного фонда (проект № 14-19-01232) в ТУСУРе. Алгоритм для решения СЛАУ разработан при поддержке гранта РФФИ 14-29-09254. Оценки ускорения выполнены в рамках государственного задания № 8.1802.2014/К Минобрнауки России.

Авторы благодарят рецензента за ценные замечания, позволившие улучшить качество статьи.

Литература

1. Григорьев А.Д. Методы вычислительной электродинамики. – М.: ФИЗМАТЛИТ, 2013. – 432 с.
2. Gazizov T.R. Analytic expressions for Mom calculation of capacitance matrix of two dimensional system of conductors and dielectrics having arbitrary oriented boundaries // Proc. of the 2001 IEEE EMC Symposium. – Montreal, Canada, 2001. – Vol. 1. – P. 151–155.
3. Новые возможности системы моделирования электромагнитной совместимости TALGAT / С.П. Куксенко, А.М. Заболоцкий, А.О. Мелкозеров, Т.Р. Газизов // Докл. Том. гос. ун-та систем упр. и радиоэлектроники. – 2015. – № 2 (36). – С. 45–50.
4. Gazizov T.R. Acceleration of Multiple Solution of a Boundary Value Problem Involving a Linear Algebraic System / T.R. Gazizov, S.P. Kuksenko, R.S. Surovtsev // Proc. of 13th Int. Conf. of Numerical Analysis and Applied Mathematics. – Rhodes, Greece, 2015. – P. 1–4.
5. Куксенко С.П. Итерационные методы решения систем линейных алгебраических уравнений с плотной матрицей / С.П. Куксенко, Т.Р. Газизов. – Томск: Том. гос. ун-т, 2007. – 208 с.
6. Sarkar T.K. Survey of numerical methods for solution of large systems of linear equations for electromagnetic field problems / T.K. Sarkar, K. Siarkiewicz, R. Stratton // IEEE Trans. Antennas Propagat. – 1981. – Vol. 29. – P. 847–856.
7. Суровцев Р.С. Аналитическая оценка вычислительных затрат на решение СЛАУ при многократном вычислении емкостной матрицы в диапазоне изменения диэлектрической проницаемости диэлектриков / Р.С. Суровцев, С.П. Куксенко, Т.Р. Газизов // Записки научных семинаров ПОМИ. – 2014. – № 428. – С. 196–207.

8. Intel MKL, коммерческая библиотека линейной алгебры [Электронный ресурс]. – Режим доступа: <http://www.intel.com>, свободный (дата обращения: 15.05.2016).

9. ATLAS, библиотека линейной алгебры [Электронный ресурс]. – Режим доступа: <http://www.netlib.org/atlas>, свободный (дата обращения: 15.05.2016).

10. Eigen, библиотека линейной алгебры [Электронный ресурс]. – Режим доступа: <http://eigen.tuxfamily.org>, свободный (дата обращения: 15.05.2016).

11. LAPACK, библиотека линейной алгебры [Электронный ресурс]. – Режим доступа: <http://www.netlib.org/lapack>, свободный (дата обращения: 15.05.2016).

12. BOOST: библиотека линейной алгебры [Электронный ресурс]. – Режим доступа: <http://www.boost.org>, свободный (дата обращения: 22.09.2015).

13. Комнатнов М.Е. Сравнение производительности математических библиотек на примере решения системы линейных алгебраических уравнений / М.Е. Комнатнов, С.П. Куксенко // Матер. докл. НТК молодых специалистов ОАО «ИСС» «Разработка, производство, испытания и эксплуатация космических аппаратов и систем», посвященная 50-летию полета в космос Ю.А. Гагарина. – Железнодорожск, 2011. – С. 396–398.

Куксенко Сергей Петрович

Канд. техн. наук, доцент каф. телевидения и управления

(ТУ) ТУСУРа

Тел.: 8 (382-2) 41-34-39

Эл. почта: ksergp@mail.ru

Суровцев Роман Сергеевич

Канд. техн. наук, ассистент каф. ТУ

Тел.: 8 (382-2) 41-34-39

Эл. почта: surovtsevrs@gmail.com

Kuksenko S.P., Surovtsev R.S.

Comparison of the computational and analytical evaluations for multiple acceleration of multiple linear system solution by block LU-decomposition

The results of the performance evaluation of noncommercial linear algebra libraries BOOST and Eigen on example of LU-decomposition are presented. It is revealed that the Eigen algorithm up to 3–53 times faster depending on variable type and Microsoft Visual C++ project configuration. Implementation of the block LU-decomposition algorithm for the multiple linear systems solution using Eigen library functions is carried out. The computational evaluations with the earlier analytical evaluations are compared. It is shown that the evaluations of acceleration are similar in type of the dependence on the number of solved linear algebraic systems, but differ on values and in type of the dependence on the number of right-hand sides.

Keywords: multiple solution, linear algebraic system, BOOST, Eigen, block LU-decomposition, analytical and computational evaluation.