АлгоВики: некоторые аспекты исследований свойств алгоритмов на примере метода Хаусхолдера*

А.В. Фролов[1], А.М. Теплов[2]

ИВМ РАН[1], НИВЦ МГУ[2]

Затронут ряд аспектов изучения свойств алгоритмов на примере исследований метода Хаусхолдера QR-разложения квадратных матриц общего вида. Обсуждаются проблемы, возникающие при вычислениях реальных производительностей, достигаемых компьютерными системами на разных программах и алгоритмах. Кроме этого, приведены результаты в области теоретического распараллеливания – две новых модификации метода Хаусхолдера с линейным критическим путём графа алгоритма.

Ключевые слова: исследование свойств алгоритмов, АлгоВики, измерение производительности вычислительных систем, критический путь графа, QR-разложение, метод Хаусхолдера.

1. Введение

Как известно [1, 7], в рамках проекта АлгоВики проводятся работы над сведением воедино информации об алгоритмах, в первую очередь, о классических. Авторы данной статьи являются также и авторами статей в АлгоВики и хотели бы здесь поделиться тем опытом, который, по мере наполнения проекта статьями, накапливается у них. В основном в этой статье будут затронуты вопросы, выявленные при работе над страницей о методе Хаусхолдера[1] [2, 3] для выполнения QR-разложения плотной квадратной матрицы. Накопленные нами сведения касаются как общих принципов работы над тестированием алгоритмов, что затронуто в разделе 2, так и специфичных свойств самого метода Хаусхолдера и его новых модификаций, приведённых в разделе 3.

2. Тестирование алгоритмов в АлгоВики

В рамках описания алгоритмов в АлгоВики представляется вполне естественным проведение тестирований и получение цифр быстродействия на разных архитектурах, с тем, чтобы рекомендации по алгоритму были основаны не только на теоретических выкладках, но и на реальных измерениях. Именно этой работой в АлгоВики занят один из авторов[2]. Как правило, программы для наполнения этого раздела берутся из каких-либо известных источников (библиотек программ), обычно вместе с тестовыми данными. Пока что тестирование проводится на вычислительных системах, доступных коллективу авторов [5], но, благодаря открытости проекта АлгоВики вполне возможно, что в будущем в этой энциклопедии свойств алгоритмов появятся и результаты тестирования на других вычислительных системах. При этом результаты будут предоставлять ценность не только для сравнения между собой разных алгоритмов применительно к одной и той же вычислительной системе, но и наоборот – для сравнения между собой вычислительных систем применительно к классам алгоритмов. Это соответствует и постепенному изменению общей направленности тестирования суперкомпьютеров в настоящее время. Как известно мировому суперкомпьютерному сообществу, замеры производительности

* Результаты, приведенные в разделе 2, получены в Московском государственном университете имени М.В. Ломоносова за счет гранта Российского научного фонда (проект №14-11-00190). Работа выполнена с использованием ресурсов суперкомпьютерного комплекса МГУ им. М.В. Ломоносова [5].

[1] В отечественной литературе, особенно советского периода, метод Хаусхолдера часто называют также методом отражений.

[2] А.М. Теплов

суперкомпьютеров, проводившиеся обычно на стандартном тесте LINPACK[1] (в том числе и на его параллельной версии), с течением времени начали расширяться. Например [16], в качестве альтернативы тесту LINPACK, имеющему вполне определённую структуру графа и потому передач информации, а также относительно выгодное для замеров соотношение между арифметическими операциями и процессами обмена, предлагают использовать и другие тесты, например HPCG[2], в котором ситуация с соотношением между арифметикой и обменами будет скорее обратная наблюдающейся у теста LINPACK. Такие замеры способны показывать вычислителям, являющимися потенциальными пользователям суперкомпьютеров, насколько эти суперкомпьютеры и их архитектура адекватно поддерживают вычислительные структуры алгоритмов нужного им типа. Однако анализ даже самых простых алгоритмов в АлгоВики показывает [1, 11, 14], насколько их структуры разнообразны. Поэтому даже наличие двух замеров по столь разным тестам, как LINPACK и HPCG, не может дать достаточно точного представления прикладнику о том, какие из суперкомпьютеров лучше поддерживают именно те структуры вычислений, которые используются в применяемых им методах.

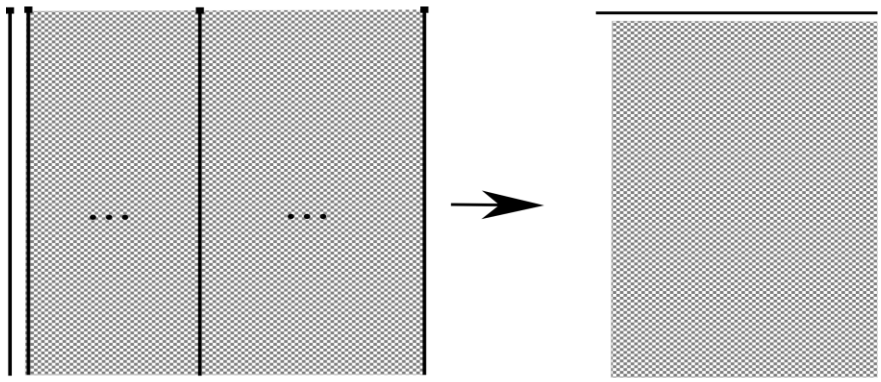

Рис. 1. Первый шаг метода Хаусхолдера.

Аналогичные работы были проделаны и по отношению к методу Хаусхолдера для вычисления QR-разложения плотной матрицы.

2.1 Особенности метода Хаусхолдера

Напомним вкратце схему метода Хаусхолдера для приведения плотной матрицы односторонними унитарными преобразованиями к треугольному виду. На рисунке 1 графически показан первый шаг. По первому столбцу подбирается матрица отражений такая, что после её умножения слева на матрицу первый столбец имеет поддиагональные нулевые элементы.

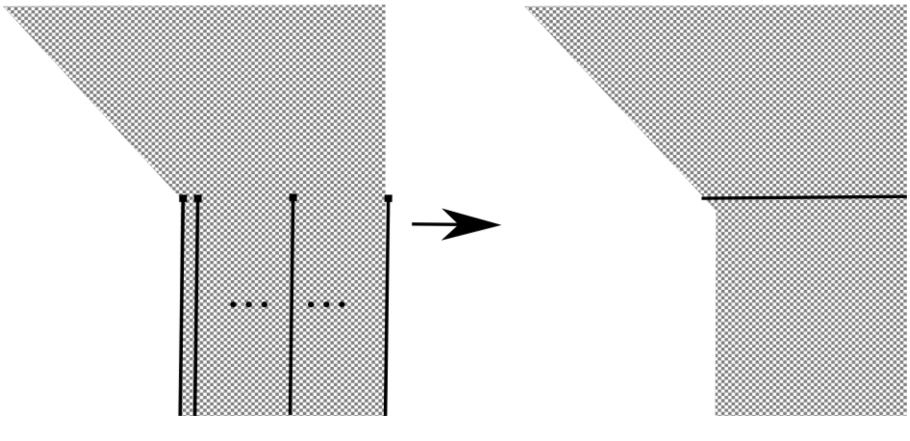

Рис. 2. *i*-й шаг метода Хаусхолдера.

[1] Этот тест основан на методе Холецкого

[2] Один из вариантов метода сопряжённых градиентов для решения СЛАУ специального вида.

На i-м шаге метода Хаусхолдера аналогичная операция проделывается над нижним правым квадратом матрицы, опирающимся на строки и столбцы с i-х по n-е. Это показано на рисунке 2.

Одно преобразование отражения (с вычислением вектора, характеризующего матрицу отражения) использует квадратичное по (n-i) число операций, а всё приведение плотной неособенной матрицы к треугольному виду – кубическое по n количество операций.

Ситуация, однако, меняется, если матрица будет иметь какой-то особенный вид. Дело в том, что в случае, если перед i-м шагом метода Хаусхолдера оказывается, что поддиагональные элементы i-го столбца и так нулевые, то преобразование отражения не выполняется. Если таких случаев мало, то количество выполняемых операций изменяется несущественно. Однако если таких столбцов оказывается большинство, то уменьшение количества операций необходимо учитывать при вычислении реального быстродействия используемой вычислительной системы. Известная из теории оценка количества операций – $4n^3/3$ – при наличии специальной структуры матрицы, ведущей к значительному числу пропусков отражений, должна быть заменена на реальное количество.

2.2 Проверка тестовых данных

Надо отметить ещё один фактор, влияющий на надёжность тестовых данных, если их применять для измерения быстродействия. Тесты для проверки программ обычно предназначаются совсем для другого. Так, когда был взят один из тестов, прилагавшийся к библиотечному QR-разложению методом Хаусхолдера, оказалось, что исходные данные представляли собой малоранговую модификацию треугольной матрицы. Ясно, что такие тесты были предназначены для проверки того, насколько верно выполняется проверка поддиагональных частей столбцов и действительно ли в нужные моменты преобразование отражения пропускается. При "вычислениях быстродействия" на таких тестах, если ошибочно исходить из максимально возможного количества операций с плавающей запятой ($4n^3/3$), получались бы фантастические цифры "быстродействия" вычислительной системы, составляющие сотни процентов от пиковой производительности.

Таким образом, при пользовании тестами исследователь, измеряющий быстродействие, не может взять тестовые данные вслепую и затем формально поделить максимальное количество операций на полученное время. Исследователь должен либо посмотреть документацию по тесту, либо, если она недоступна, проверить постфактум после запуска полученные при расчётах результаты. Так, в случае метода Хаусхолдера обычно в документации (по программе, а не по тестовым данным) указывается, в каком конечном виде хранятся данные о матрицах элементарных отражений. Поэтому после замера времени, необходимого на выполнение самого метода, следует дописать программу проверки соответствующих структур.

В случае же, если такая проверка тоже недоступна, рекомендуется для измерения производительности взять другие тестовые данные. Так было сделано при исследованиях программ, реализующих метод Хаусхолдера для QR-разложения матриц. На странице АлгоВики, посвящённой этому методу, измерения соответствуют тестам, где отражения выполняются на всех его шагах.

3. Критический путь графа алгоритма в методе Хаусхолдера

В АлгоВики исследование параллельных свойств алгоритмов включает в себя и определение таких их параметров, как критический путь графа алгоритма и ширина наискорейшей ярусно-параллельной формы. Конечно, поскольку считать всерьёз условия так называемой концепции неограниченного параллелизма [4] выполнимыми на реальных архитектурах нельзя, эти характеристики не полностью характеризуют возможности распараллеливания алгоритма. Тем не менее они остаются важнейшими в плане первичной оценки таких возможностей. Известно [6, 7, 14], например, что при распараллеливании простейшей прогонки возникают большие трудности. Поэтому интересными являются даже частично неустойчивые [12, 13, 17], но параллельные её замены, и при этом исследования могут возвращаться и к, казалось, забытым методам [6, 15]. Более того, различные методы уменьшения критического пути графа, если помнить о целях проекта АлгоВики, необходимо должны в нём присутствовать – хотя бы как примеры.

При исследовании метода Хаусхолдера для QR-разложения плотных матриц перед исследователем встали, помимо других, более простых, вопросы: во-первых, насколько критический путь его графа меньше или больше критического пути в других методах QR-разложения и, во-вторых, нет ли способа, оставаясь в рамках использования преобразований Хаусхолдера, модифицировать метод с тем, чтобы уменьшить критический путь его графа.

Сразу сделаем оговорки. Во-первых, при сравнении с другими методами рассматривались только такие, которые применимы к проведению QR-разложения матрицы без дополнительных условий. Это сразу исключило из рассмотрения метод, который основывается на разложении матрицы A^*A методом Холецкого в произведение R^*R и дальнейшим вычислением $Q = AR^{-1}$: он требует невырожденности матрицы A (см. [2], Теорема 27.2) и не годится для квадратной матрицы общего вида. Аналогично из рассмотрения исключён и классические варианты метода ортогонализации, поскольку при той же вырожденности матрицы A получения полного набора ортогональных столбцов в матрице Q в нём не достичь. После этого для проведения сравнений вместе с методом Хаусхолдера остался только метод Гивенса[1]: оба они работают и в условиях вырожденности[2]. Во-вторых, при проведении модификации рассматривались только "точечные" варианты метода Хаусхолдера, а блочные варианты обойдены стороной. Это связано как с общим подходом к описаниям в АлгоВики, где блочные методы рассматриваются отдельно, так и с тем, что их совместный анализ пока что был бы слишком самонадеянной, по мнению авторов, попыткой.

3.1 Классический метод Хаусхолдера и метод Гивенса

Если посмотреть на описание метода Хаусхолдера в его классическом виде [2], то очевидно, что, как бы мы его ни видоизменяли, с помощью таких модификаций никак нельзя достичь того, чтобы критический путь графа стал меньше, чем линейная функция от размера матрицы. Дело в том, что для выполнения каждого следующего шага просто необходимо, чтобы уже были выполнены все предыдущие. Это хорошо видно на рисунке 2, где необходимым условием для выполнения шага являются нули слева от текущего столбца. Уже это даёт нижнюю оценку, пропорциональную количеству шагов.

Если же рассмотреть структуру каждого i-го шага, то видно, что для вычисления вектора отражения нужно предварительно найти евклидову норму вектора размерности $n\text{-}i\text{+}1$, а для выполнения преобразования – скалярные произведения вектора отражения с другими векторами такой же размерности. Если находить эти нормы и скалярные произведения классически последовательно, то на каждом шаге получается линейно зависимое от $n\text{-}i\text{+}1$ выражение для пути графа, и если просуммировать все шаги, то в результате квадратное выражение от n, что и записано на странице АлгоВики. Используя приёмы типа сдваивания, можно уменьшить критический путь графа каждого отражения до логарифмического, а критический путь графа всего алгоритма до $O(n\ log_2 n)$.

Однако далее уменьшать критический путь графа, оставаясь в рамках классического метода Хаусхолдера, невозможно. Это невыгодно отличает его от другого метода QR-разложения – метода Гивенса, называемого также методом вращений. Действительно, на рисунках 3 и 4 показано, как можно привести матрицу к треугольному виду, выполняя независимые вращения параллельно друг другу, за линейное число параллельных шагов. Жирными точками на рисунках обозначены ведущие элементы, сплошными линиями – преобразуемые строки.

Интересно, что для метода Гивенса линейная оценка критического пути снизу тоже имеет место – по тем же причинам, что и в методе Хаусхолдера: пока слева от какого-либо поддиагонального элемента преобразуемой матрицы остаются другие ненулевые элементы, нельзя приступать к его собственному "обнулению". Однако у метода Гивенса эта теоретическая оценка критического пути графа снизу достижима на практике. В связи с этим возникает вопрос: а можно ли как-то преобразовать метод Хаусхолдера, не изменяя его сути, заключённой в вы-

[1] В отечественной литературе его называют также методом вращений

[2] Естественно, в случае вырожденности матрицы A матрица R будет содержать нули на диагонали, но искомое разложение $A=QR$ может быть найдено и в методе Гивенса, и в методе Хаусхолдера, причём в их классических вариантах.

полнении преобразований отражения, так, чтобы и для критического пути его графа получилась линейная по размеру матрицы оценка?

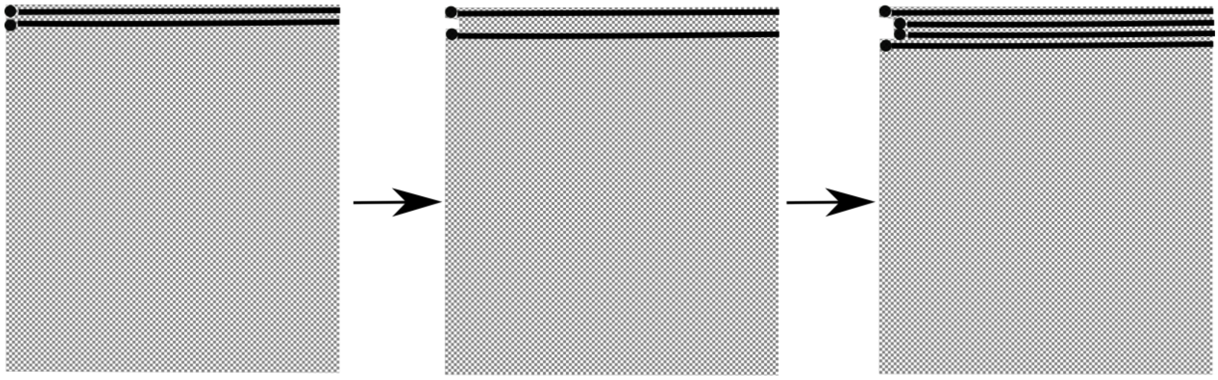

Рис. 3. Первые шаги метода Гивенса.

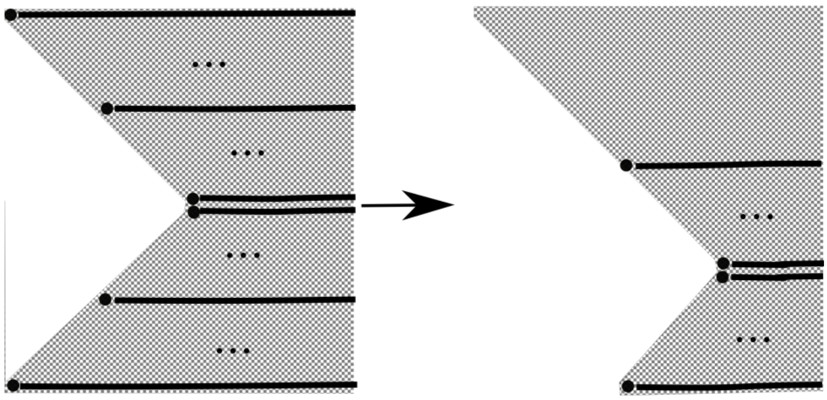

Рис. 4. Параллельно исполняемые шаги метода Гивенса.

Прежде чем модифицировать классическую схему коренным образом, рассмотрим способ сокращения критического пути у одного преобразования Хаусхолдера.

3.2 Вычисления скалярных произведений "сокращённым порядком"

Вернёмся к схеме одного шага метода Хаусхолдера, который представляет собой одно преобразование отражения. Сначала нам надо найти само преобразование, а потом уже выполнить его. Матрица отражений для подпространства размерности *k (k=n-i+1)* находится в форме

$$U = E - \frac{1}{\gamma} v v^* \tag{1}$$

где параметры (скаляр γ и вектор v) вычисляются из условия приведения вектора s (он здесь соответствует первому из столбцов не приведённого к треугольной форме остатка матрицы) к вектору, коллинеарному единичному орту e_1. Согласно [2], если

$$(s, s) = 0 \tag{2}$$

то $\gamma = ½$, $v = e_1$.

Если же

$$(s, s) \neq 0 \tag{3}$$

то

$$v = \frac{1}{\sqrt{(s,s)}} s \pm e_1 , \qquad \gamma = 1 + \frac{|s_1|}{\sqrt{(s,s)}} \tag{4}$$

где знак в выражении берётся тот же, что у s_1 (а при нулевом значении любой, например плюс).

Если теперь вектор z (в качестве которого выступает при преобразовании любой другой столбец не приведённого к треугольному виду остатка матрицы) подвергнуть найденному преобразованию отражения U, то результат будет равен вектору

$$Uz = z - \frac{(z,v)}{\gamma} v = z - \beta v \tag{5}$$

и коэффициент при v равен

$$\beta = \frac{(z,v)}{\gamma} = \frac{\frac{1}{\sqrt{(s,s)}}(z,s) \pm z_1}{1 + \frac{|s_1|}{\sqrt{(s,s)}}} = \frac{(z,s) \pm \sqrt{(s,s)} z_1}{\sqrt{(s,s)} + |s_1|} \tag{6}$$

а модифицирующая z добавка равна

$$\beta v = \frac{(z,s) \pm \sqrt{(s,s)} z_1}{\sqrt{(s,s)} + |s_1|} \left(\frac{1}{\sqrt{(s,s)}} s \pm e_1 \right) = \frac{(z,s) \pm \sqrt{(s,s)} z_1}{(s,s) + |s_1| \sqrt{(s,s)}} \left(s \pm \sqrt{(s,s)} e_1 \right) \tag{7}$$

При этом для первой компоненты вектора получается

$$(Uz)_1 = z_1 - \frac{(z,s) \pm \sqrt{(s,s)} z_1}{(s,s) + |s_1| \sqrt{(s,s)}} \left(s_1 \pm \sqrt{(s,s)} \right) \tag{8.1}$$

а для остальных

$$(Uz)_j = z_j - \frac{(z,s) \pm \sqrt{(s,s)} z_1}{(s,s) + |s_1| \sqrt{(s,s)}} s_j \tag{8.2}$$

Таким образом, если вычислять скалярное произведение (z,s) параллельно вычислению (s,s), то само умножение матрицы отражений на вектор z уже не потребует таких дополнительных вычислений, у которых критический путь был бы зависящим от k. Поэтому даже без особой модификации метода Хаусхолдера этот приём даёт почти двукратный выигрыш в длине критического пути его графа, используется при этом приём сдваивания или нет. Однако на этом останавливаться рано. Сократить критический путь можно ещё, используя простейшие свойства унитарных матриц.

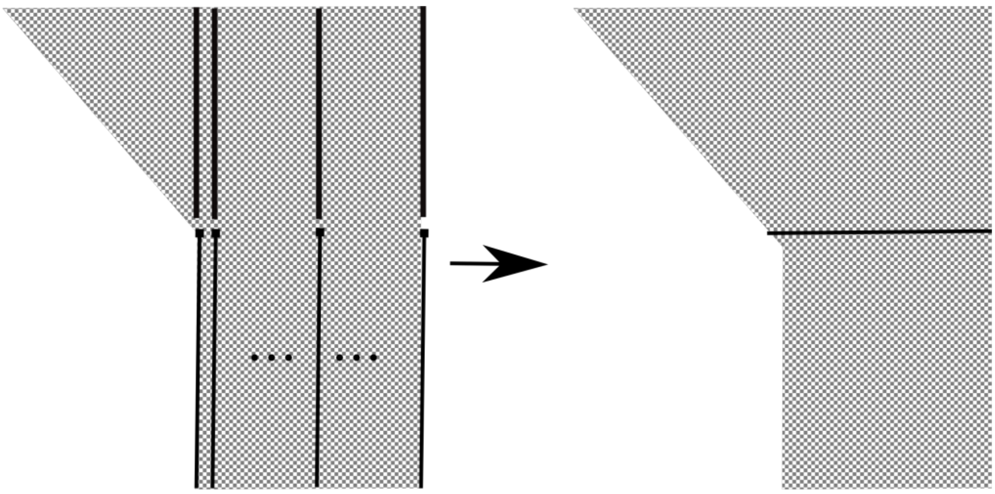

Рис. 5. Шаг метода Хаусхолдера с отмеченными векторами.

3.3 Модификация метода Хаусхолдера с использованием свойств матрицы Грама

В предыдущей части из формул (5), (7) видно, что преобразование отражения (Хаусхолдера) одного шага метода потребует (в параллельном режиме) конечное число арифметических операций, если скалярные произведения, фигурирующие в (7), будут как-то предвычислены. На рисунке 5 слева на промежуточной матрице показан набор тех векторов, скалярные произведения которых нужно вычислить перед выполнением преобразования. При этом внизу, начиная с выделенных точками элементов – те части столбцов, которые изменяются в ходе шага-преобразования и которые должны быть учтены при вычислении скалярных произведений, а вверху, обозначенные жирными линиями, – те части столбцов, которые в данном шаге-преобразовании меняться не будут и в скалярных произведениях могут в классической схеме метода Хаусхолдера не учитываться.

Однако учёт верхних частей столбцов может помочь для предвычисления скалярных произведений, нужных для определения коэффициентов в формулах (6-7). Если считать, что влияние ошибок округления на унитарность матриц отражения в процессе приведения исходной матрицы к треугольному виду пренебрежимо мало, то скалярные произведения полных столбцов должны сохраняться:

$$(Ux, Uy) = (x, U^*Uy) = (x, y) \tag{9}$$

При этом для двух любых столбцов z^k и z^m нужное для вычисления параметров отражения скалярное произведение их "низа"

$$(z^k, z^m)_H = \sum_{j=i}^{n} z_j^k \overline{z_j^m} \tag{10}$$

может быть вычислено из полного скалярного произведения столбцов

$$(z^k, z^m) = \sum_{j=1}^{n} z_j^k \overline{z_j^m} \tag{11}$$

и скалярного произведения "верха" столбцов

$$(z^k, z^m)_B = \sum_{j=1}^{i-1} z_j^k \overline{z_j^m} \tag{12}$$

с помощью простого вычитания:

$$(z^k, z^m)_H = \left(z^k, z^m\right) - (z^k, z^m)_B \tag{13}$$

Ещё одно важное наблюдение: первые элементы "верха" столбцов фиксируются сразу после первого преобразования отражения, вторые после второго и т.д. последовательно.

Исходя из этих наблюдений, можно предложить такую схему QR-разложения, опирающуюся на преобразования отражения, в котором будет достигнута линейная оценка критического пути графа алгоритма. Эта схема предполагает, что в начале процесса должна быть вычислена матрица Грама с элементами $g_{kj} = \left(z^k, z^j\right)$ для системы всех столбцов z^k исходной матрицы.

Теперь на i-м шаге метода в формулах (1-7) вместо (s, s) используются элементы g_{ii}, а вместо каждого из (z, s) элементы g_{ji}. После выполнения шага элементы матрицы Грама g_{kj} *(j>i, j≤k)* перевычисляются вычитанием из их предыдущего значения произведения k-го и j-го элементов i-й строки. После этого перевычисления процесс готов к следующему шагу.

Вне зависимости о того, будут ли первичные вычисления матрицы Грама использовать сдваивание или нет, главный член в формулах критического пути графа алгоритма в такой схеме получается линейный. Цена этой линейности в терминах ресурсов составляет дополнительные $2n^3/3$ операций и $n(n+1)/2$ дополнительной памяти (а с учётом того, что надо всё выполнять в режиме накопления, то вдвое больше). Но главная цена линейности новой схемы, скорее всего[1], состоит в её худших параметрах устойчивости в сравнении с классической. Впрочем, и классический метод Хаусхолдера способен дать такую же хорошую устойчивость, как метод Гивенса, только если при вычислениях применяется режим накопления.

Можно, однако, не использовать допущение унитарности преобразований, дающее сохранение скалярных произведений. Это даёт ещё одну схему с линейным критическим путём графа алгоритма, правда, совсем уж непрактичную из-за гигантского суммарного объёма вычислений.

[1] Авторы ещё не проводили анализ новой схемы на влияние ошибок округления. По первому впечатлению предположение унитарности применяемых преобразований, не выполняемое при реальных расчётах на вычислительной системе, может привести к худшей устойчивости, чем классический метод Хаусхолдера. Возможно, что до самой конференции удастся провести такой анализ и ответить на вопрос об устойчивости.

3.4 Модификация метода Хаусхолдера с использованием матриц Грама в подпространствах

Пусть у нас известны скалярные произведения трёх векторов *a, b, c*: *(a,b), (a,a), (b,b), (a,c), (b,c), (c,c).* Пусть также вектора a и b преобразуются по формулам

$$a' = a - \sigma c, \quad b' = b - \tau c \tag{14}$$

Тогда скалярные произведения преобразованных векторов вычисляются через скалярные произведения "старых" по формулам

$$(a', a') = (a, a) + \sigma^2(c, c) - \sigma\overline{(a, c)} - \overline{\sigma}(a, c) \tag{15.1}$$

$$(b', b') = (b, b) + \tau^2(c, c) - \tau\overline{(b, c)} - \overline{\tau}(b, c) \tag{15.2}$$

$$(a', b') = (a, b) + \sigma\tau(c, c) - \sigma\overline{(b, c)} - \overline{\tau}(a, c) \tag{15.3}$$

которые можно вычислить за конечное количество шагов. Эти формулы дают ещё один способ QR-разложения плотных матриц методом Хаусхолдера без промежуточного вычисления длинных скалярных произведений, нужных при вычислении параметров преобразований отражения, и потому с линейным критическим путём графа алгоритма.

Этот способ сводится к тому, что на каждом шаге выполняются вычисления и перевычисления матриц Грама в подпространствах. Каждая из матриц Грама в i-м подпространстве G^i хранит скалярные произведения векторов, составленных из компонент столбцов с i-й по n-ю. На старте схемы вычисляются все такие матрицы Грама. Затем на каждом i-м шаге для вычисления параметров отражения по формулам (1-7) используются элементы матрицы G^i, а элементы матриц Грама G^k $(k>i)$ перевычисляются по формулам (15.1-15.3). Нетрудно видеть, что эта схема также выполнима за линейное время, но при этом для её выполнения требуется выполнить $O(n^4)$ арифметических операций, и хранить при этом нужно $O(n^3)$ данных. Указанные особенности делают возможность применения схемы на практике столь умозрительной, что проведение анализа ошибок округления в ней вряд ли имеет смысл. Однако можно заметить: применение многошаговых формул (15) столь схоже с процессами ортогонализации, что процесс перевычислений матриц Грама вряд ли будет устойчивым.

Если вернуться к вопросу о том, насколько значимо нахождение последних двух схем, использующих разными способами матрицу Грама, то авторы находят возможным рассмотреть две стороны этого вопроса. Во-первых, тот же метод Стоуна [7,9,17] стал хрестоматийным для тех, кто начинает изучение параллельных алгоритмов: несмотря на свою неустойчивость, он в условиях точных вычислений даёт лучшие показатели по длине критического пути. Во-вторых, и это отмечено и здесь, и на странице АлгоВики, метод Хаусхолдера по многим параметрам уступает методу Гивенса, но, благодаря большей опоре на базовые функции типа скалярного произведения, преобладает в библиотеках программ, в том числе для суперкомпьютеров. Поэтому на эту тему, во всяком случае до изучения влияния ошибок округления, пока можно только привести стандартную фразу, что всё рассудит время.

4. Заключение

Выделим основные результаты, полученные авторами при исследовании метода Хаусхолдера для выполнения QR-разложения.

Выделены потенциальные источники ошибок при тестировании быстродействия программ на различных вычислительных системах. Даны рекомендации, как их избежать.

Для метода Хаусхолдера предложены схемы, теоретически сокращающие критический путь графа алгоритма, в том числе две новые схемы с линейной сложностью. Для условий точных вычислений схемы дают то же разложение, что и классический метод Хаусхолдера.

Следует подчеркнуть, что новые схемы в очередной раз получены при исследовании свойств классических методов для размещения в энциклопедии АлгоВики. Специалисты в каких-либо методах, уделившие внимание рассказу об их структурах в рамках этой электронной энциклопедии, вполне могут в рамках выполнения задачи описания посмотреть на эти алгоритмы по-новому, что, не исключено, тоже даст им какие-то новые исследовательские идеи.

Литература

1. Антонов А.С., Воеводин Вад.В., Воеводин Вл.В., Теплов А.М., Фролов А.В.. Первая версия Открытой энциклопедии свойств алгоритмов // Вестник УГАТУ. Серия управление, вычислительная техника и информатика. Том 19, N 2(68), 2015., С.150-159
2. Воеводин В.В. Вычислительные основы линейной алгебры. М.: Наука, 1977.
3. Воеводин В.В., Кузнецов Ю.А. Матрицы и вычисления. М.: Наука, 1984.
4. Воеводин В.В. Математические основы параллельных вычислений// М.: Изд. Моск. ун-та, 1991. 345 с.
5. Воеводин Вл., Жуматий С., Соболев С., Антонов А., Брызгалов П., Никитенко Д., Стефанов К., Воеводин Вад. Практика суперкомпьютера «Ломоносов» // Открытые системы, 2012, N 7, С. 36-39.
6. Ильин В.П., Кузнецов Ю.И. Трехдиагональные матрицы и их приложения. М.: Наука. Главная редакция физико-математической литературы, 1985г. ,208 с.
7. Открытая энциклопедия свойств алгоритмов. URL: http://algowiki-project.org (дата обращения: 15.04.2017).
8. Фаддеев Д.К., Фаддеева В.Н. Вычислительные основы линейной алгебры. М.-Л.: Физматгиз, 1963.
9. Фаддеева В.Н., Фаддеев Д.К. Параллельные вычисления в линейной алгебре 1,2. // Кибернетика, 1977. №6. С. 28-40; 1982. №3. С. 18-31.
10. Фролов А.В.. Принципы построения и описание языка Сигма. Препринт ОВМ АН N 236. М.: ОВМ АН СССР, 1989.
11. Фролов А.В., Воеводин Вад.В., Коньшин И.Н., Теплов А.М. Исследование структурных свойств алгоритма разложения Холецкого: от давно известных фактов до новых выводов // Параллельные вычислительные технологии (ПаВТ'2015): труды международной научной конференции (31 марта - 2 апреля 2015 г., г. Екатеринбург). Челябинск: Издательский центр ЮУрГУ, 2015. С. 320-331.
12. Фролов А.В. Ещё один метод распараллеливания прогонки с использованием ассоциативности операций // Суперкомпьютерные дни в России: Труды международной конференции (28-29 сентября 2015 г., г. Москва). – М.: Изд-во МГУ, 2015. с. 151-162
13. Фролов А.В. Использование последовательно-параллельного метода для распараллеливания алгоритмов с ассоциативными операциями // Суперкомпьютерные дни в России: Труды международной конференции (28-29 сентября 2015 г., г. Москва). – М.: Изд-во МГУ, 2015. с. 176-184
14. Фролов А.В., Антонов А.С., Воеводин Вл.В., Теплов А.М. Сопоставление разных методов решения одной задачи по методике проекта Algowiki // Параллельные вычислительные технологии (ПаВТ’2016): труды международной научной конференции (г. Архангельск, 28 марта – 1 апреля 2016 г.). Челябинск: Издательский центр ЮУрГУ, 2016. С. 347-360.
15. Фролов А.В. Нециклическая редукция - незаслуженно забытый метод? // Параллельные вычислительные технологии (ПаВТ’2016): труды международной научной конференции (г. Архангельск, 28 марта – 1 апреля 2016 г.). Челябинск: Издательский центр ЮУрГУ, 2016. С. 800.
16. Jack Dongarra, Mark Gates, Azzam Haidar, Yulu Jia, Khairul Kabir, Piotr Luszczek, and Stanimire Tomov. HPC Programming on Intel Many-Integrated-Core Hardware with MAGMA Port to Xeon Phi // Scientific Programming. Article Number: 502593.
17. Stone H.S. Parallel Tridiagonal Equation Solvers // ACM Trans. on Math. Software, Vol. 1, No. 4 (Dec. 1975), P. 289-307.

AlgoWiki: some aspects of algorithm properties investigations with Householder QR-decomposition studying as an example

A.V. Frolov[1], A.M. Teplov[2]

INM RAS[1], SRCC MSU[2]

Some aspects of algorithm properties investigations in AlgoWiki are concerned in the article. Householder QR-decomposition studying is an example. Efficiency calculations peculiarities are discussed. This article presents two new modifications of Householder QR decomposition with linear critical path.

Keywords: algorithm properties investigations, AlgoWiki, QR decomposition, Householder method, critical path of graph.

References

1. Antonov A.S., Voevodin Vad.V., Voevodin Vl.V., Teplov A.M., Frolov A.V.. Pervaya versiya Otkrytoy entsiklopedii svoystv algoritmov [First Version of Algorithms' Properties' Open Encyclopedia] // Vestnik UGATU. Seriya upravlenie, vychislitel'naya tekhnika i informatika [Bulletin of UGATU. Series: Control, Compruting Technique & Informatics]. Tom 19, N 2(68), 2015., S.150-159.
2. Voevodin V.V. Vychislitel'nye osnovy lineynoy algebry [Computing Basics of Linear Algebra]. M.: Nauka, 1977.
3. Voevodin V.V., Kuznetsov Yu.A. Matritsy i vychisleniya [Matrices & Computing]. M.: Nauka, 1984.
4. Voevodin V.V. Matematicheskie osnovy parallel'nykh vychisleniy [Mathematics' Basics of Parallel Computing]. M.: Izd. Mosk. un-ta, 1991. 345 s.
5. Voevodin Vl., Zhumatiy S., Sobolev S., Antonov A., Bryzgalov P., Nikitenko D., Stefanov K., Voevodin Vad. Praktika superkomp'yutera «Lomonosov» [Lomonosov Supercomputer Practice] // Otkrytye sistemy [Open Systems], 2012, №7, S. 36-39.
6. Il'in V.P., Kuznetsov Yu.I. Trekhdiagonal'nye matritsy i ikh prilozheniya [Tridiagonal Matrices & Applications]. M.: Nauka. Glavnaya redaktsiya fiziko-matematicheskoy literatury, 1985g., 208 s.
7. Otkrytaya entsiklopediya svoystv algoritmov [Algorithms' Properties' Open Encyclopedia]. URL: http://algowiki-project.org (accessed: 15.04.2017).
8. Faddeev D.K., Faddeeva V.N. Vychislitel'nye osnovy lineynoy algebry [Computing Basics of Linear Algebra]. M.-L.: Fizmatgiz, 1963.
9. Faddeeva V.N., Faddeev D.K. Parallel'nye vychisleniya v lineynoy algebre [Concurrent computing in Linear Algebra] 1,2. // Kibernetika [Cybernetics], 1977. №6. S. 28-40; 1982. №3. S. 18-31.
10. Frolov A.V.. Printsipy postroeniya i opisanie yazyka Sigma [Sigma Language Design Principles & Description]. Preprint OVM AN №236. M.: OVM AN SSSR, 1989.
11. Frolov A.V., Voevodin Vad.V., Kon'shin I.N., Teplov A.M. Issledovanie strukturnykh svoystv algoritma razlozheniya Kholetskogo: ot davno izvestnykh faktov do novykh vyvodov [Cholesky Algorithm Structure Properties' Investigation: from Old Facts to New Conclusions]// Parallel'nye vychislitel'nye tekhnologii (PaVT'2015): trudy mezhdunarodnoy nauchnoy konferentsii (31 marta - 2 aprelya 2015 g., g. Ekaterinburg) [Parallel Computational Technologies (PCT'2015): Proceedings of the International Scientific Conference (Ekaterinburg, Russia, March, 31 – April, 2, 2015)]]. Chelyabinsk, Publishing of the South Ural State University, 2015. P. 320–331.

12. Frolov A.V. Eshchye odin metod rasparallelivaniya progonki s ispol'zovaniem assotsiativnosti operatsiy [Yet another Tomas Algorithm Parallelizing Method with Operathions Associativity Using]// Superkomp'yuternye dni v Rossii: Trudy mezhdunarodnoy konferentsii (28-29 sentyabrya 2015 g., g. Moskva) [Russian Supercomputer Days: Proceedings of the International Scientific Conference (Moscow, Russia, September, 28-29, 2015]. Moscow, MSU Publishing, 2015. P. 151-162

13. Frolov A.V. Ispol'zovanie posledovatel'no-parallel'nogo metoda dlya rasparallelivaniya algoritmov s assotsiativnymi operatsiyami [Sequential-Parallel Method's Using for Algorithms Containing Associative Operations Parallelizing]// Superkomp'yuternye dni v Rossii: Trudy mezhdunarodnoy konferentsii (28-29 sentyabrya 2015 g., g. Moskva) [Russian Supercomputer Days: Proceedings of the International Scientific Conference (Moscow, Russia, September, 28-29, 2015]. Moscow, MSU Publishing, 2015. P. 176-184

14. Frolov A.V., Antonov A.S., Voevodin Vl.V., Teplov A.M. Sopostavleniye raznyh metodov resheniya odnoj zadachi po metodike proekta Algowiki [One problem solving different methods' comparison according to the criteria of the Algowiki project] // Parallel'nye vychislitel'nye tekhnologii (PaVT'2016): trudy mezhdunarodnoy nauchnoy konferentsii (28 marta – 1 aprelya 2016 g., g. Arkhangelsk) [Parallel Computational Technologies (PCT'2016): Proceedings of the International Scientific Conference (Arkhangelsk, Russia, March, 28 – April, 1, 2016)]. Chelyabinsk, Publishing of the South Ural State University, 2016. P. 347–360.

15. Frolov A.V. Neciklicheskaya redukciya – nezasluzhenno zabytyy metod? [Noncyclic reduction is undeservedly forgotten method?] // Parallel'nye vychislitel'nye tekhnologii (PaVT'2016): trudy mezhdunarodnoy nauchnoy konferentsii (28 marta – 1 aprelya 2016 g., g. Arkhangelsk) [Parallel Computational Technologies (PCT'2016): Proceedings of the International Scientific Conference (Arkhangelsk, Russia, March, 28 – April, 1, 2016)]. Chelyabinsk, Publishing of the South Ural State University, 2016. P. 800.

16. Jack Dongarra, Mark Gates, Azzam Haidar, Yulu Jia, Khairul Kabir, Piotr Luszczek, and Stanimire Tomov. HPC Programming on Intel Many-Integrated-Core Hardware with MAGMA Port to Xeon Phi // Scientific Programming. Article Number: 502593.

17. Stone H.S. Parallel Tridiagonal Equation Solvers // ACM Trans. on Math. Software, Vol. 1, No. 4 (Dec. 1975), P. 289-307.