AlgoWiki: о некоторых характеристиках новых алгоритмов*

А.В. Фролов[1], А.С. Антонов[2], Н.А. Фролов[3]

Федеральное государственное бюджетное учреждение науки Институт вычислительной математики Российской академии наук[1], Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный университет имени М.В.Ломоносова»[2], Федеральное государственное автономное образовательное учреждение высшего образования «Московский физико-технический институт (государственный университет)»[3]

Как уже неоднократно отмечалось в работах авторов за предыдущие несколько лет, исследования классических методов в рамках подготовки их описаний для энциклопедии AlgoWiki периодически дают идеи новых алгоритмов, в чём-либо превосходящих классические. Эти идеи озвучиваются в соответствующих статьях. Однако полное исследование нового алгоритма в рамках работы над описаниями для AlgoWiki невозможно, поскольку выходит далеко за рамки намеченного плана и требует значительного времени. В данной статье сделана попытка наметить некоторые общие для всех новых идей характеристики, которые подлежат исследованиям в первую очередь.

Ключевые слова: AlgoWiki, вычислительные методы линейной алгебры, параллельные структуры.

1. Введение

После начала работ над AlgoWiki [1,2] ряд авторов статей этой открытой энциклопедии свойств алгоритмов не только переносит в Wiki-формат информацию, найденную в учебниках и статьях и касающуюся очередного описываемого алгоритма или метода. Для того, чтобы статьи-описания соответствовали требованиям современности, желательно также провести самостоятельное исследование того, насколько эта сумма знаний соответствует требованиям суперкомпьютерной эпохи: нельзя ли найти что-то либо новое [3–7], либо хорошо забытое старое [8], ставшее теперь более интересным. Авторами настоящей статьи в ряде предыдущих работ были выделены эти либо новые, либо забытые старые аспекты, касающиеся решения тех же задач, для которых предназначены классические алгоритмы, и полученные с помощью соотнесения данных задач с возможным распараллеливанием алгоритмов [9].

В данной статье излагается сводный обзор полученных авторами за последние годы новых идей по распараллеливанию. Здесь, однако, все они рассматриваются с одной позиции. Авторы пытаются дать ответ на простой вопрос. Если некоторая новая (забытая старая) идея даёт в каком-то плане лучшие возможности для распараллеливания, то какие стадии изучения должно пройти её исследование?

2. Критерии и улучшение возможностей для распараллеливания

Одной из важных характеристик, дающих исследователям оценки по возможностям распараллеливания, является критический путь графа алгоритма. Следует, однако, помнить, что прямой зависимости "меньший критический путь – лучше распараллеливание" нет, поскольку в реальном мире не выполняются условия т.н. "концепции неограниченного параллелизма" [9]. Тем не менее, в качестве первого критерия, с целью уменьшения которого появляются новые идеи распараллеливания, критический путь вполне подходит, если не забывать и о других.

* Результаты, приведенные в разделах 2.2 и 3, получены в Московском государственном университете имени М.В. Ломоносова за счет гранта Российского научного фонда (проект №14-11-00190).

Скажем, простая оценка количества арифметических операций в алгоритме является довольно важной, хотя уже давно известно и неоднократно озвучено на конференциях, что содержащий меньшее количество операций алгоритм не обязательно быстрее выполняется. Однако в том случае, если количество арифметических операций в сравниваемых алгоритмах отличается на порядки, то практически при любой возможности распараллеливания более "тяжёлый" алгоритм вряд ли можно распараллелить так, что он будет исполняться быстрее.

В том случае, если два описанных выше параметра слабо отличаются для сравниваемых алгоритмов, важными становятся уже более тонкие структурные их характеристики. Скажем, наличие или отсутствие дисбаланса между шириной разных ярусов, локальность и т.п. Однако не менее важными являются вычислительные характеристики, вообще напрямую не связанные со структурными свойствами алгоритма. Это – влияние ошибок округления на ошибки при нахождении решения.

Разберём влияние разных характеристик на конкретных примерах.

2.1 Решение трёхдиагональных СЛАУ

Классическим методом решения трёхдиагональных систем линейных алгебраических уравнений является прогонка [10], которая в её изначальной форме не поддаётся распараллеливанию (за исключением встречной прогонки, которая, однако, всего-навсего удваивает ширину ярусно-параллельной формы графа и потому, в основном, способна только эксплуатировать многоядерность нынешних процессоров [11]). Для многократного же распараллеливания прогонка непригодна. В связи с этим уже "на заре суперкомпьютерной эры", когда начинали использовать параллельность обработки данных, были придуманы такие методы, как схема Стона [12, 13] и другие методы вроде простой и циклической редукции [10,13]. Схема Стона, основанная на приёме сдваивания и использующая ассоциативность операции матричного умножения, с логарифмической длиной критического пути графа имела наименьший коэффициент и в 1973–1975 гг. ещё казалась наилучшей, однако достаточно быстро была обнаружена её неустойчивость к ошибкам округления. При этом сама схема была такова, что на стадии, аналогичной прямому ходу прогонки, неустойчивость была неустранимой из схемы. Поэтому со временем наиболее популярной из параллельных замен прогонки стала циклическая редукция, также имеющая логарифмическую длину критического пути графа алгоритма, хотя и с несколько б**о**льшим коэффициентом.

При анализе данных алгоритмов, однако, выяснилось, что, несмотря на все свои преимущества, циклическая редукция, при полном использовании необходимого количества процессоров, использует большинство из них только на первом и на последнем своих шагах. Это ведёт к низкой реальной эффективности. Кроме того, коммуникационной схемой, наилучшим образом пригодной для циклической редукции, оказывается гиперкуб. Относительно него были оптимистические предположения в 80-90-х гг. XX века, однако по мере миниатюризации интегральных схем и приближения к концу закона Мура они ушли в прошлое. Распространённые же коммуникационные схемы подходят для циклической редукции плохо.

Поэтому в статьях [3–5, 8] по результатам этого анализа были предложены разные заменители прогонке и циклической редукции – все на основе последовательно-параллельной схемы. Первым вариантом [3] была схема с последовательно-параллельным [4] использованием ассоциативности умножения матриц, в которой, в отличие от схемы Стона [12, 13], на достаточно длинных последовательных ветвях можно было использовать такой хорошо известный вычислителям метод, как нормализация. Подобный подход давал лучшую устойчивость, чем схема Стона, однако при детальном изучении всё же оказалось, что устойчивость нового метода гарантируется для более узкой области данных, чем у алгоритмов из класса прогонок. Поэтому после изучения литературы и старых алгоритмов, было предложено возвращение к такому старому способу, как нециклическая редукция [8], а именно – к её последовательно-параллельной реализации.

При этом оказалось, что анализ устойчивости простой (нециклической) редукции давно проведён в [10], и заново его делать не нужно. Последней частью анализа, после которого можно переходить к практическим рекомендациям по данному алгоритму, было сравнение по

загруженности систем разными методами. При этом предположения, сделанные сначала в [5], оказались на поверку не выполненными, и по результатам расчётов для проверки адаптации на малоядерных процессорах [11] оказалось, что для элементарных частей последовательно-параллельной схемы нециклической редукции лучше всего должны подойти встречные прогонки, более полно использующие ресурсы ядер процессоров.

В настоящее время, конечно, у ряда исследователей, избалованных работой на мощных системах, есть определённое пренебрежение к проблеме распараллеливания решения трёхдиагональных СЛАУ: на небольших системах можно обойтись и прогонкой, а для больших систем уже всё равно начинает сказываться общая неустойчивость, связанная с ростом обусловленности. Однако общение с решающими реальные задачи на стендовой секции конференции по материалам [8] показало, что для многих исследователей задача "распараллеливания прогонки" остаётся актуальной, ибо не у всех хватает вычислительных мощностей. При этом подзадачи с трёхдиагональными СЛАУ довольно часты в приложениях, так что актуальности проблема вовсе не утратила.

Поэтому для тех, кто занимается численными расчётами на параллельных вычислительных системах с использованием подзадач, использующих решение СЛАУ с трёхдиагональными матрицами, таким образом, в случае, когда нет априорной информации об элементах этих матриц, авторы считают наилучшим вариантом последовательно-параллельную схему нециклической редукции, причём с использованием схемы встречных прогонок в каждой из последовательных частей. При наличии же априорной информации можно использовать разные монотонные прогонки по рекомендациям [10].

2.2 QR-разложения матриц

Следующей проблемой, изученной в ряду других задач, было нахождение QR-разложений матриц общего вида. После изучения метода Гивенса (вращений), для которого достаточно просто находится линейная от размера матриц зависимость длины критического пути графа алгоритма[1], был исследован более популярный в пакетах и библиотеках программ метод Хаусхолдера (отражений).

Было отмечено, что для стандартного метода Хаусхолдера длина критического пути графа алгоритма зависит от размера матрицы квадратично, и даже при использовании методов сдваивания для вычисления скалярных произведений[2] – как $O(n \log_2 n)$. Линейность же (в зависимости от размера матриц) длины критического пути графа алгоритма для метода Хаусхолдера в его стандартном виде недоступна.

При общем сравнении методов видно, что при сравнении по количеству операций формально картина противоположна, у метода Гивенса коэффициент при n^3 больше, чем у метода Хаусхолдера (2 против 4/3). То, что в пакетах программ решения задач линейной алгебры гораздо популярнее метод Хаусхолдера, может быть связано именно с этим фактором. При этом, однако, для аналогичных методу Гивенса характеристик устойчивости в методе Хаусхолдера должен для вычисления всех скалярных произведений использоваться режим накопления [14]. Поэтому основная доля операций в методе Хаусхолдера должна быть выполнена в режиме двойной точности, что практически съедает его преимущество. В тоже время, лучшая распараллеливаемость метода Гивенса, казалось бы, должна дать ему преимущество в выборе метода для формирования пакета прикладных программ. Однако, возможно, что в глазах конструкторов пакетов метод Хаусхолдера имеет то преимущество, что почти весь составлен из стандартных базовых операций (BLAS) – скалярных произведений и взвешенных суммирований векторов.

[1] Как и в [9], критическим путём в [2] и здесь называем самый длинный из путей в графе алгоритма, где все вершины соответствуют однократному исполнению арифметической операции (для вычислительных методов обычно операции с плавающей запятой). Напомним, что длина критического пути естественным образом ограничивает снизу возможную высоту ярусно-параллельной формы графа, поэтому является теоретической оценкой времени, нужного для выполнения алгоритма в концепции неограниченного параллелизма.

[2] нежелательном с точки зрения равномерной загруженности устройств

При исследовании был поставлен и такой теоретический вопрос: может ли быть достигнут линейный (в зависимости от размера матрицы) критический путь графа алгоритма, основанного именно на преобразованиях Хаусхолдера? Оказалось, что таких способов даже два, и они описаны в [6].

Однако один из этих алгоритмов, основанный на предвычислении и последующих перевычислениях всех "частных" матриц Грама столбцов матрицы, явно не подлежит реализации: количество операций в нём зависит от размера матрицы как $O(n^4)$, т.е. на порядок превышает количество операций стандартного метода Хаусхолдера. В таких условиях естественно, что сокращение длины критического пути не может иметь никакого значения.

Другой алгоритм основан на предвычислении и корректировке только одной матрицы Грама столбцов исходной матрицы, а также на свойствах унитарных/ортогональных преобразований сохранять для образов скалярные произведения прообразов. При этом увеличение общего количества операций, однако, становится равным количеству операций в методе Гивенса. Естественно, что и наличие режима накопления в этой версии метода Хаусхолдера для обеспечения устойчивости необходимо, но вовсе не достаточно: использование свойств унитарных/ортогональных преобразований в условиях действия ошибок округления может вполне привести к общей неустойчивости этого варианта метода.

Таким образом, если нужно распараллелить метод Хаусхолдера, то проверка устойчивости (если не получится теоретически, то, например, практическими сравнениями со стандартным разложением) этого варианта нужна. Однако пока что проверку варианта на устойчивость в планы работы включать преждевременно. Это связано с тем, что сначала следует проверить на эффективность (и сравнить со стандартным методом Хаусхолдера) распараллеленный вариант метода Гивенса, обладающего той же оценкой возмущения без использования режима двойной точности (для накопления). До этого улучшение параллельных характеристик, основанное на преобразованиях Хаусхолдера (отражений), не имеет практического смысла.

2.3 QR-алгоритм

Пока что последним из методов, где при исследовании структуры графа алгоритма появились новые идеи для распараллеливания, является QR-алгоритм [15–20], а если более точно – QR-итерации после приведения матрицы к хессенберговому виду. Дело в том, что в применяемых в пакетах стратегиях выбора сдвигов (*правило Фрэнсиса*[1] или, в случае симметричных матриц, *стратегия Уилкинсона*) [15] их вычисление зависит от правого нижнего угла матрицы, полученной от итераций с предыдущими сдвигами. В общем случае применения таких стратегий замечено, что количество итераций линейно зависит от размера матрицы. Каждая итерация имеет критический путь, также линейно зависящий от размера[2]. Поэтому оценка критического пути всех итераций вместе составляет $O(n^2)$. Это ограничение снизу пытаются преодолеть различными способами [19, 20], в том числе как увеличением размера нижнего правого угла очередного блока[3], так и предвычислением с помощью оценок следующих сдвигов.

В [7] авторами предложена идея использовать для вычисления сдвигов не правый нижний, а левый верхний угол. Это позволило бы вычислять совместно сразу много итераций. Однако для такой модификации нужно убедиться, что количество итераций в такой стратегии останется линейно зависимым от размера матрицы. При этом было бы даже не так важно, если бы коэффициент при n увеличился, поскольку взамен получилось бы массовое параллельное выполнение многих итераций с разными сдвигами[4] и суммарный линейный (по n) критический путь графа всего алгоритма.

Поэтому, в отличие от предыдущих примеров, для QR-итераций следует в первую очередь выполнить исследование влияния переноса вычисления сдвигов "вверх" на сходимость

[1] Оставим в стороне частные случаи специальных матриц, для которых при выборе сдвига по правилу Фрэнсиса не сходится QR-итерация, и периодически приходится выполнять сдвиг по особым формулам.
[2] далее размер матрицы будем обозначать буквой n.
[3] Однако сильное увеличение размера, естественно, невозможно. Это ограничивает количество совместно выполняемых итераций.
[4] Влияние повторения сдвигов уже исследовано в цитируемых работах.

итераций. В связи с этим в планах в первую очередь экспериментальные запуски со сравнением на тестах старых способов вычисления сдвигов с новым. В первую очередь это следует сделать у версии QR-итераций для несимметричных матриц.

В случае, если количество итераций с новой стратегией выбора сдвигов останется линейным по размерам матриц, это предварительное исследование позволит приступить на следующем этапе работ уже к разработке параллельной схемы QR-итераций со сдвигами, с полным учётом всех аспектов от простой независимости вычислений друг от друга до планов по оптимальному распределению данных по узлам вычислительных систем. Однако при неудаче на предыдущем этапе, если количество итераций будет больше линейного, проведение этой работы будет неоправданной затратой рабочего (и, возможно, машинного) времени.

Для нахождения собственных значений симметричных матриц в настоящее время QR-итерации уступают по скорости вычислений методу «разделяй и властвуй». Однако это не означает, что скорость сходимости QR-итераций для нового выбора сдвигов неинтересна в симметричном случае. Дело в том, что даже метод «разделяй и властвуй» после разбиения матрицы на блоки достаточно малого размера использует в них не что иное, как QR-итерации. Поэтому это направление исследования сходимости QR-итераций с новой стратегией выбора сдвигов тоже должно быть интересным, как на матрицах небольшой размерности, так и (в лучшем случае) для большой размерности[1].

3. Об отображении новых идей в AlgoWiki

Как видно из вышеизложенного, далеко не все из новых идей по распараллеливанию решения той или иной задачи проходят отбраковку перед практической реализацией. Означает ли это, что какие-либо из перечисленных идей не должны в принципе найти своё отражение на страницах описаний энциклопедии AlgoWiki?

На этот вопрос мы должны дать отрицательный ответ. Действительно, во-первых, сами по себе идеи, с помощью которых улучшаются какие-либо характеристики алгоритма (уменьшается длина критического пути его графа, повышается локальность вычислений, регулярность дуг графа и т.п.) достаточно ценны. То, что в одном случае может дать ухудшение других важных характеристик алгоритма (устойчивости, скорости сходимости и т.п.), в другом случае может не сопровождаться таким отрицательным эффектом. В качестве такого примера может служить та же схема Стона: в той части, которая является заменой прямому ходу прогонки, она неустойчива, но в части, заменяющей обратный ход, её устойчивость не хуже, чем у заменяемого алгоритма[2]. Поэтому, несмотря на свою неустойчивость, схема Стона, хоть и не используется в практике, описана общим планом на соответствующих страницах AlgoWiki. Во-вторых, описания отбракованных алгоритмов и причин, по которым их не применяют, нужны читателям энциклопедии алгоритмов для того, чтобы элементарно не «изобрести велосипед», к тому же ещё и «сломанный».

В связи с этим авторы собираются со временем дополнить AlgoWiki описаниями всех алгоритмов, о которых уже было сказано в предыдущих разделах статьи, как было сделано с уже упомянутой схемой Стона. При этом, однако, в зависимости от того, насколько подробно было произведено их исследование, будет определён уровень описания (метод, алгоритм) и степень его детализации в энциклопедии.

4. Заключение

Как видно на примерах, появление новой идеи для распараллеливания метода (а точнее, для предложения изменения схемы решения задачи на ту, которая лучше поддаётся распараллеливанию) не обязательно сразу ведёт к попыткам её практической реализации.

[1] Если таким образом получится ускорить QR-итерации для симметричных матриц, то они, возможно, снова смогут составить конкуренцию методу «разделяй и властвуй». Впрочем, до проведения исследований на скорость сходимости при новом способе выбора сдвигов об этом ещё рано говорить.

[2] Аналогично можно сказать и про уже описанную выше последовательно-параллельную схему замены прогонки, также основанную на ассоциативности умножения матриц.

Прежде всего, отбраковка перед практическим программированием может быть по двум основным характеристикам:

а) слишком много дополнительных вычислений в новой схеме по сравнению со стандартной (как в одной из модификаций метода Хаусхолдера, где общий объём вычислений для новой схемы стал четвёртой степени вместо кубической зависимости у классической версии);

б) новая схема оказывается явно неустойчивой (как в случае последовательно-параллельной замены прогонки, основанной на ассоциативности операции умножения матриц).

В этих случаях дальнейшая работа по развитию конкретной идеи не стоит затраченного на неё времени и не проводится. Тем не менее, бывает, что сразу после получения новой идеи неясно, устойчива она или нет. Тогда для получения ясности приходится проводить специальное исследование. Если теоретических оценок провести не удаётся, то вполне можно оценить устойчивость через эксперименты над тестовыми данными.

В случаях, когда дополнительных вычислений в новой схеме немного, и при этом с её устойчивостью всё в порядке, в дело вступают следующие характеристики схемы:

в) регулярность графа алгоритма и локальность вычислений [1, 2], как это было для последовательно-параллельных вариантов разных схем;

г) скорость сходимости в случаях итерационных методов, как в случае с QR-итерациями.

Наконец, важным триггером в деле продолжения работ является наличие или отсутствие хорошей альтернативы, как, например, метод Гивенса (вращений) является альтернативой методам, основанным на преобразованиях Хаусхолдера (отражений). В этом случае представляется важным понять, почему эта хорошая альтернатива менее популярна в пакетах программ. Если таковые причины окажутся, например, архаичными, то тогда следует реализовать, несомненно, лучшую альтернативу, не оставаясь в рамках группы алгоритмов.

Таким образом, при наличии небольших ресурсов исследователь «на распутье» должен вначале оценить перспективы выбираемого пути, и сделать это на основе перечисленных характеристик новых идей.

И, в заключение, напоминаем известную[1] формулу "Отрицательный результат – тоже результат". Исследователь, столкнувшийся с тем, что идея нового алгоритма по каким-либо причинам не приводит к хорошей реализации, все же должен опубликовать как идею, так и причину, помешавшую ее хорошему воплощению, это сэкономит уйму рабочего времени у его последователей.

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. Открытая энциклопедия свойств алгоритмов. URL: http://algowiki-project.org (дата обращения: 30.03.2018).
3. 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.
4. 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.
5. 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.

[1] приписываемую Нильсу Бору

6. Фролов А.В., Теплов А.М. АлгоВики: некоторые аспекты исследований свойств алгоритмов на примере метода Хаусхолдера // Суперкомпьютерные дни в России: Труды международной конференции (25–26 сентября 2017 г., г. Москва). – М.: Изд-во МГУ, 2017. – с.500–510.
7. Фролов А.В., Антонов А.С. AlgoWiki: опыт исследования ряда алгоритмов // Параллельные вычислительные технологии – XII международная конференция, ПаВТ'2018, г. Ростов-на-Дону, 2–6 апреля 2018 г. Короткие статьи и описания плакатов. Челябинск: Издательский центр ЮУрГУ, 2018. – с.366–375.
8. 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.
9. Воеводин В.В. Математические основы параллельных вычислений. М.: Изд. Моск. ун-та, 1991. 345 с.
10. Ильин В.П., Кузнецов Ю.И. Трехдиагональные матрицы и их приложения. М.: Наука. Главная редакция физико-математической литературы, 1985г., 208 с.
11. Фролов Н.А., Фролов А.В. Экспериментальные исследования влияния степени локальности алгоритмов на их быстродействие на примере решения трёхдиагональных СЛАУ // Тезисы 59й научной конференции МФТИ (21–26 ноября 2016 г., гг. Москва–Долгопрудный). URL: http://conf59.mipt.ru/static/reports_pdf/1490.pdf (дата обращения: 30.03.2018).
12. Stone H.S. An Efficient Parallel Algorithm for the Solution of a Tridiagonal Linear System of Equations // J. ACM, Vol. 20, No. 1 (Jan. 1973), P. 27-38. DOI: 10.1145/321738.321741.
13. Stone H.S. Parallel Tridiagonal Equation Solvers // ACM Trans. on Math. Software, Vol. 1, No. 4 (Dec. 1975), P. 289-307. DOI: 10.1145/355656.355657.
14. Воеводин В.В. Вычислительные основы линейной алгебры. М.: Наука, 1977. 304 с.
15. Деммель Дж. Вычислительная линейная алгебра. Теория и приложения. Пер. с англ. – М.: Мир, 2001. 430 с.
16. Кублановская В.Н. О некоторых алгоритмах для решения полной проблемы собственных значений // ЖВМиМФ, 1961. Т. 1. № 4. С. 555–570.
17. Francis J.G.F. The QR Tranformation A Unitary Analogue to the LR Transformation—Part 1 // The Computer Journal, 1961, vol. 4, no. 3, pp. 265–271. DOI: 10.1093/comjnl/4.3.265.
18. Francis J.G.F. The QR Tranformation—Part 2 // The Computer Journal, 1962, vol. 4, no. 4, pp. 332–345. DOI: 10.1093/comjnl/4.4.332.
19. Braman K., Byers R., Mathias R. The multishift QR algorithm, I: Maintaining well-focused shifts and level 3 performance // SIAM J. Matrix Anal. Appl., 23(4): 929–947, 2002. DOI: 10.1137/s0895479801384573.
20. Braman K., Byers R., Mathias R. The multishift QR algorithm, II: Aggressive early deflation // SIAM J. Matrix Anal. Appl., 23(4):948–973, 2002. DOI: 10.1137/s0895479801384585.