УДК 519.61

Ю. Н. Базилевич
Приднепровская государственная академия строительства и архитектуры, г. Днепропетровск

НОВЫЕ ВЫЧИСЛИТЕЛЬНЫЕ АЛГОРИТМЫ В ЗАДАЧАХ ТОЧНОЙ ДЕКОМПОЗИЦИИ

Разработаны алгоритмы, позволяющие эффективно находить общее инвариантное подпространство нескольких матриц. Рассмотрено расширение области применения программы, предназначенной для нахождения общего решения большой разреженной системы линейных однородных алгебраических уравнений.

Розроблено алгоритми, що дозволяють ефективно знаходити спільний інваріантний підпростір декількох матриць. Розглянуто розширення сфери застосування програми, призначеної для знаходження спільного розв'язку великої розрідженої системи лінійних однорідних алгебраїчних рівнянь.

The algorithms allowing finding effectively the common invariant subspace of several matrixes are developed. Increasing opportunities of the program designed for finding of the total decision of the big rarefied system of the linear uniform algebraic equations is considered.

Ключевые слова: алгоритм, декомпозиция, матрицы, общее решение, эффективность.

Введение. Точная декомпозиция [1; 2] — это методы, позволяющие либо привести данную систему уравнений к нескольким подсистемам, либо установить, что при выбранном классе преобразований такое приведение невозможно. Разделение на независимые подсистемы соответствует приведению нескольких матриц коэффициентов к блочно-диагональному виду одним преобразованием. Иерархическая декомпозиция (последовательная, вертикальная декомпозиция, приводимость или редукция) соответствует приведению матриц к блочно-треугольному виду. Подсистемы уравнений, соответствующие блокам матриц, стоящим на главной диагонали, позволяют исследовать устойчивость решений исходной системы уравнений как при разделении на независимые подсистемы, так и при иерархической декомпозиции.

Постановка задачи

Для решения задач декомпозиции необходим алгоритм нахождения общего решения большой разреженной системы линейных однородных алгебраических уравнений. Таких алгоритмов сравнительно мало. В [3, с. 130] приведены универсальные алгоритмы svd и minfit для нахождения общего решения. Специальный алгоритм SLAU5 для решения такой задачи в случае большой сильно разреженной матрицы коэффициентов изложен в [1]. Для решения задач иерархической декомпозиции необходим ещё алгоритм ALG нахождения общего инвариантного подпространства нескольких матриц [4].

Алгоритм SLAU5 (и соответствующая компьютерная программа) показал достаточно высокую эффективность. Представляет интерес расширение его области применения.

Второй алгоритм недостаточно эффективен. Применяя его для решения практических задач, пришлось использовать двойную точность. Возникли также проблемы при попытке выполнить приближённую декомпозицию, аналогичную выполненной в [1].

В настоящей работе решаются следующие задачи.

- Повышение точности и быстродействия существующего алгоритма нахождения общего инвариантного подпространства нескольких матриц.
- Расширения области применения алгоритма SLAU5 нахождения общего решения большой сильно разреженной системы линейных однородных алгебраических уравнений.

Метод решения и анализ полученных результатов

1. В существующем алгоритме нахождения общего инвариантного подпространства нескольких матриц при составлении алгебры, порождённой данными матрицами, производятся многократные проверки линейной зависимости матриц. Проверяется совокупность «предполагаемого базиса» и дополнительной матрицы.

Причина недостаточной эффективности алгоритма связана со следующим: несмотря на то что понятие линейной независимости качественное (зависимы или независимы векторы), с вычислительной точки зрения эта независимость может проявляться более или менее чётко. Например, два вектора на плоскости, расположенные под острым углом друг к другу, будут линейно независимыми, если этот угол не равен нулю. Но когда угол сравнительно мал, компьютерная программа, работающая с ограниченной точностью, может посчитать эти векторы линейно зависимыми, поэтому отсутствие ортогонализации и нормировки элементов «предполагаемого базиса» приводит к накоплению погрешностей и возможным ошибкам. Здесь

$n \times n$-матрица рассматривается как вектор длиной n^2. Сама проверка выполняется с помощью программы SLAU5, которая превращает первые части векторов «предполагаемого базиса» в единичные векторы длины m, где m — это количество векторов данного «базиса». Для этих векторов свойство линейной независимости будет проявляться более чётко. Далее такой базис будем называть улучшенным.

В алгоритм следует внести следующее изменение: после формирования очередного варианта «предполагаемого базиса» и установления его линейной независимости с помощью программы SLAU5 заменить эти векторы на «улучшенные», использованные программой как промежуточный результат. Такое изменение даст также некоторое увеличение быстродействия, поскольку операции с векторами, содержащими много нулей, будут выполняться быстрее.

Заметно сократить время работы алгоритма можно, изменив порядок пересмотра всех возможных произведений матриц. Обычная организация циклов здесь не эффективна потому, что при добавлении нового элемента к базису процесс надо начинать заново с новой длиной цикла. Итак, нужно организовать перебор всех возможных сочетаний индексов i и j от 1 до некоторого значения r, величина которого заранее не известна.

Изменённый алгоритм составления алгебры матриц, порожденной исходными матрицами $\{B_k\}$, следующий:

а) к матрицам системы $\{B_k\}$ добавляем матрицу $B_0 = E$;

б) выбираем улучшенный базис во множестве матриц, новые матрицы базиса обозначим $W_1, W_2, \ldots, W_\mu$;

в) вычисляем произведения матриц W_iW_j. Если какое-либо из них не является линейной комбинацией матриц $\{W_k\}$, то принимаем его новым элементом множества $\{W_k\}$. Подробнее:

(i) полагаем $r := \mu$; $m := 1$;

(ii) для значений $i = 1$, $j = 1$ выполняем пункт (iv), после чего переходим к пункту (vi);

(iii) полагаем $i := m$; затем для всех значений j от 1 до $m - 1$ выполняем пункт (iv), после чего переходим к пункту (v);

(iv) вычисляем $Z := W_iW_j$, проверяем линейную зависимость матриц $W_1, W_2, \ldots, W_r, Z$. Если они линейно независимы, полагаем $r := r + 1$, $W_r := Z$ и заменяем базис множества $\{W_k\}$ на улучшенный;

(v) $j := m$; затем для всех значений i от 1 до m повторяем выполнение пункта (iv);

(vi) $m := m + 1$, если $m \le n$, возвращаемся к пункту (iii).

В результате получим базис $W_1, W_2, \ldots, W_r$ алгебры с единицей, порождённой исходными матрицами $\{B_k\}$.

Далее (рисунок) показаны номера вычисляемых в соответствии с вышеприведенным алгоритмом произведений матриц W_iW_j и соответствующие номера индексов i и j.

		j		
		1	2	3
i	1	**1**	**3**	**7**
	2	**2**	**4**	**8**
	3	**5**	**6**	**9**
	4	**10**	**…**	

Последовательность вычисления произведений

Если программу нахождения общего инвариантного подпространства нескольких матриц составить на языке алгебраического программного комплекса GAP [5], то основные задачи — составление алгебры, порождённой данными матрицами, и вычисление её радикала будут записаны всего в несколько строчек. Но при этом возникают другие проблемы: вычисления будут произведены в арифметике рациональных чисел (не вещественных). Это означает, что нельзя ввести насчитанные заранее матрицы. Нужно непосредственно в программе на языке GAP задать исходные данные с точностью до трёх (например) значащих цифр и трактовать их как дробные числа; затем в этой же программе выполнить формирование матриц, соответствующих исследуемой системе, т.е. избежать громоздких вычислений всё равно не удастся. Выявление систем, близких к приводимым, таким способом невозможно. Тем не менее такой принципиально новый подход представляет интерес.

2. Можно решить неоднородную систему уравнений с помощью программы нахождения общего решения однородной системы (SLAU5 или аналогичной). Для этого вместо системы уравнений $A\mathbf{x} = \mathbf{b}$ составляем систему уравнений

$$A_\text{р}\, \mathbf{y} = \mathbf{0}, \tag{1}$$

где $A_\text{р}$ — расширенная матрица коэффициентов, т. е. такая, что к матрице коэффициентов A справа присоединён вектор свободных членов $\mathbf{b}$; $\mathbf{y}$ — вектор, имеющий размерность на единицу большую, чем вектор $\mathbf{x}$. Новая система эквивалентна предыдущей в случае, когда последний элемент вектора $\mathbf{y}$ равен минус единице: $y_{m+1} = -1$. Поэтому, получив базис во множестве решений системы (1), выбираем такой вектор базиса, у которого $y_{m+1} \ne 0$, и умножаем этот вектор на такое число, при котором последний элемент станет равным минус единице. Предыдущие m элементов вектора $\mathbf{y}$ превратятся в частное решение исходной системы уравнений, т. е. вектор **x**. Если среди векторов базиса нет такого, что $y_{m+1} \neq 0$, то неоднородная система не имеет решения.

Общее решение неоднородной системы уравнений, как известно, равняется сумме частного решения неоднородной системы и общего решения соответствующей однородной системы уравнений.

В традиционном случае, когда матрица A квадратная и невырожденная, при решении уравнений (1) будет получен один вектор базиса, и условие $y_{m+1} \neq 0$ будет выполнено. Умножив вектор базиса на число $-1/y_{m+1}$, получим вектор с решением исходной системы уравнений. Таким образом, программа, предназначенная для работы с большой разреженной системой уравнений при решении задач декомпозиции, может использоваться и для других задач, например, для применения метода конечных элементов. Основные достоинства исходной программы сохраняются.

Для нахождения базиса подпространства, являющегося ортогональным дополнением к данному подпространству, нужно расположить в строках массива A векторы базиса исходного подпространства (либо такие векторы, что исходное подпространство является их линейной оболочкой). Затем — обратиться к программе нахождения общего решения системы линейных однородных алгебраических уравнений $A\mathbf{x} = \mathbf{0}$. Базисом искомого подпространства будет полученное общее решение системы уравнений. Действительно, запись $A\mathbf{x} = \mathbf{0}$ означает, что скалярное произведение любой строки на вектор **x** равняется нулю, т.е. вектор-столбец, составленный из элементов строки матрицы, ортогонален вектору **x**. Общее решение — это и есть все те векторы, которые ортогональны всем строкам матрицы одновременно.

Выводы

Поставленные задачи решены.

Дальнейшее развитие работ по декомпозиции целесообразно вести по следующим направлениям:

- выявление систем, близких к таким, для которых иерархическая декомпозиция возможна;
- обобщение метода приведения матриц к блочно-треугольному виду на системы уравнений, содержащие прямоугольные матрицы;
- практическое применение методов декомпозиции для новых прикладных задач.

Библиографические ссылки

1. **Базилевич, Ю. Н.** Численные методы декомпозиции в линейных задачах механики [Текст] / Ю. Н. Базилевич. — К.: Наук. думка, 1987. — 156 с.
2. **Базилевич, Ю. Н.** Точная декомпозиция линейных систем [Электронный ресурс] / Ю. Н. Базилевич // Исследовано в России. — Режим доступа: http://www.sci-journal.ru/articles/2006/018.pdf.
3. **Уилкинсон, Дж. Х.** Справочник алгоритмов на языке АЛГОЛ. Линейная алгебра [Текст] / Дж. Х. Уилкинсон, С. Райнш. — М.: Машиностроение, 1976. — 389 с.
4. **Базилевич, Ю. Н.** Решение задачи иерархической декомпозиции линейных математических моделей механических систем [Текст] / Ю. Н. Базилевич, М. Л. Коротенко, И. В. Швец // Техн. механика. — 2003. — №1. — С. 135 — 140.
5. Сайт Украинской группы пользователей GAP [Электронный ресурс]. — Режим доступа: http://www.gap-system.org/ukrgap/

Надійшла до редколегії 09.06.2015