Ю.В. Цыганова

# АЛГОРИТМИЧЕСКОЕ ВЫЧИСЛЕНИЕ ПРОИЗВОДНЫХ В МАТРИЧНЫХ ОРТОГОНАЛЬНЫХ ПРЕОБРАЗОВАНИЯХ

(Ульяновский государственный университет)

## Введение

В докладе представлен подход к построению методов вычисления производных в матричных ортогональных преобразованиях. Ортогональные преобразования в силу своих улучшенных вычислительных свойств широко используются при решении различных задач вычислительной линейной алгебры [1]. В теории калмановской фильтрации ортогональные преобразования применяются для робастного (устойчивого к ошибкам машинного округления) вычисления решения матричного разностного уравнения Риккати [2].

В свою очередь, потребность в дифференцировании матричных соотношений для вычисления значений производных элементов параметризованных матриц возникает в задачах вычислительной математики, математической физики, теории управления и др. *Параметризованной матрицей* $A(\theta)$ назовём прямоугольную матрицу размера $m \times n$, элементы которой являются дифференцируемыми функциями по параметру $\theta$.

В настоящее время наиболее распространены три способа вычисления производных [3]: символьное (аналитическое) дифференцирование, численное дифференцирование и алгоритмическое (автоматическое) дифференцирование.

Символьное дифференцирование позволяет получить точные аналитические формулы для производных элементов параметризованной матрицы, однако даже для матрицы небольших размеров такой подход требует существенных вычислительных затрат и не подходит для решения задач в режиме реального времени. При численном дифференцировании точность вычислений существенно зависит от многих факторов, например, от величины шага выбранного алгоритма. В отличие от двух предыдущих способов, алгоритмическое (автоматическое) дифференцирование позволяет вычислить не выражения для производных и не их табличное приближение, а численные значения производных при данных значениях аргументов функции. Для этого требуется знание выражения для функции или хотя бы компьютерной программы для её вычисления.

В связи с этим, целью доклада является описание подхода к построению методов алгоритмического вычисления производных матричных ортогональных преобразований параметризованных матриц.

Рассмотрим пример классического $QR$ преобразования. Построим алгоритм нахождения в заданной точке $\theta$ значений производных $\partial r_{ij} / \partial \theta$ элементов верхней треугольной матрицы $R(\theta)$, полученной в результате матричного ортогонального преобразования $A(\theta) = QR(\theta)$, где $A(\theta)$ – параметризованная по $\theta$ прямоугольная матрица, $Q$ – матрица ортогонального преобразования.

## 1. Метод вычисления производных в QR преобразовании

Сформулируем теоретический результат [4]:

Пусть задана прямоугольная матрица $A(\theta)$ размера $(s+k)\times(s+l)$, где $s>0$, $k,l\geq 0$, и элементы матрицы являются дифференцируемыми функциями по скалярному параметру $\theta$. Рассмотрим матричное ортогональное преобразование $A(\theta)=QR(\theta)$, представленное в блочном виде:

$$\begin{bmatrix}(A_{11})_{s\times s} & (A_{12})_{s\times l}\\ (A_{21})_{k\times s} & (A_{22})_{k\times l}\end{bmatrix}=Q\begin{bmatrix}(R_{11})_{s\times s} & (R_{12})_{s\times l}\\ O_{k\times s} & (R_{22})_{k\times l}\end{bmatrix}, \quad (1)$$

где $O_{k\times s}$ – нулевой матричный блок.

Применим то же ортогональное преобразование к матрице производных $A'_\theta$:

$$Q^T\begin{bmatrix}(A'_{11})_{s\times s} & (A'_{12})_{s\times l}\\ (A'_{21})_{k\times s} & (A'_{22})_{k\times l}\end{bmatrix}=\begin{bmatrix}X_{s\times s} & N_{s\times l}\\ Y_{k\times s} & V_{k\times l}\end{bmatrix}, \quad (2)$$

где $X$, $N$, $Y$, $V$ – соответствующие матричные блоки.

Тогда при известной матрице производных $A'_\theta$ и заданном значении параметра $\theta=\hat{\theta}$, значения производных элементов верхней треугольной матрицы $R'_\theta$ можно вычислить по формулам:

$$(R'_{11})_{\hat{\theta}}=\left(\overline{L}^T+D+\overline{U}\right)R_{11}, \quad (3)$$

$$(R'_{12})_{\hat{\theta}}=\left(\overline{L}^T-\overline{L}\right)R_{12}+R_{11}^{-T}Y^TR_{22}+N, \quad (4)$$

где $\overline{L}$, $D$ и $\overline{U}$ – соответственно строго нижняя треугольная, диагональная и строго верхняя треугольная части матричного произведения $X\cdot R_{11}^{-1}$.

Строгое математическое доказательство содержится в [4].

Теперь запишем вычислительный алгоритм:

### Алгоритм 1 вычисления производных в *QR* преобразовании:

**Вход:** Значение параметра $\theta=\hat{\theta}$, матрица $A(\hat{\theta})$ и матрица производных $A'_{\hat{\theta}}$.

**Выполнить:** ортогональное преобразование $A=QR$.

**Найти:** произведение $Q^T\cdot A'_{\hat{\theta}}$ и представить его в блочном виде (2).

**Вычислить:** произведение $X\cdot R_{11}^{-1}$ и разбить его на части $\overline{L}$, $D$ и $\overline{U}$.

**Найти:** матрицы $(R'_{11})_{\hat{\theta}}$ и $(R'_{12})_{\theta=\hat{\theta}}$ по выражениям (3) и (4).

**Выход:** значения производных матричных блоков $R_{11}$ и $R_{12}$.

Замечание. В представленном алгоритме не вычисляются значения производных матричного блока $R_{22}$, поскольку такой случай редко используется для решения практических задач.

## 2. Численный пример

Проведем вычислительные эксперименты по исследованию точности представленного алгоритма 1 и времени его выполнения. Рассмотрим два типа

тестовых матриц $A(\theta)$ размера $m \times n$ $(m,n = 2,10,100,1000)$, заданных своими элементами [5]:

- **Тип 1:** $a_{ij} = \dfrac{1}{1+\theta \cdot \left(\frac{(i-1)\,j}{m}\right)^4}$, где $i = 1,\ldots,m$, $j = 1,\ldots,n$; $\hat{\theta} = 66$.
- **Тип 2:** $a_{ij} = \theta \cdot (RAND - 0.5)$, где $i = 1,\ldots,m$, $j = 1,\ldots,n$; $\hat{\theta} = 100$,

*RAND* – случайное число из равномерного распределения.

Погрешность вычислений будем оценивать по формуле:

$$e = \left\| (A^T \cdot A)'_{\hat{\theta}} - (R^T \cdot R)'_{\hat{\theta}} \right\|_{\infty}.$$

Исходный код алгоритма 1 реализован в программной среде MATLAB R2010a. Эксперименты проведены на персональном компьютере с характеристиками: Intel Core 2 Quad Q6600 @ 2.4 GHz, 4 GB RAM, MS Windows 7 Ultimate. Результаты экспериментов представлены в таблицах 1 и 2. Видно, что для вычисления производных в ортогональном преобразовании матрицы размера 1000×1000 потребовалось менее 0.8 сек, при этом погрешность вычислений не превысила $5.2 \cdot 10^{-10}$. Таким образом, полученные результаты подтверждают эффективность построенного алгоритма 1 для использования в задачах реального времени и для систем больших размеров.

Таблица 1. Временные затраты на выполнение алгоритма 1 (сек).

| | **Тип 1** | | | | **Тип 2** | | | |
|---|---|---|---|---|---|---|---|---|
| (m,n) | 2 | 10 | 100 | 1000 | 2 | 10 | 100 | 1000 |
| 2 | 0.0059 | 0.4531 | 0.0002 | 0.0073 | 0.0060 | 0.0065 | 0.0001 | 0.0070 |
| 10 | 0.0001 | 0.0003 | 0.0066 | 0.0262 | 0.0001 | 0.0001 | 0.0022 | 0.0006 |
| 100 | 0.0002 | 0.0015 | 0.0142 | 0.0241 | 0.0002 | 0.0028 | 0.0107 | 0.0241 |
| 1000 | 0.0256 | 0.0691 | 0.4726 | 0.6051 | 0.0290 | 0.0717 | 0.5069 | 0.7553 |

Таблица 2. Погрешность вычислений по алгоритму 1.

| | **Тип 1** | | | | **Тип 2** | | | |
|---|---|---|---|---|---|---|---|---|
| (m,n) | 2 | 10 | 100 | 1000 | 2 | 10 | 100 | 1000 |
| 2 | 2.8e-19 | 2.2e-18 | 2.6e-17 | 2.7e-16 | 3.5e-14 | 4.7e-14 | 4.3e-13 | 5.6e-12 |
| 10 | 3.9e-18 | 7.6e-18 | 5.9e-17 | 1.0e-15 | 8.5e-14 | 1.6e-13 | 5.6e-12 | 5.9e-11 |
| 100 | 6.2e-17 | 1.2e-16 | 1.8e-15 | 2.5e-14 | 5.8e-13 | 1.3e-12 | 1.2e-11 | 1.6e-10 |
| 1000 | 1.7e-15 | 2.2e-15 | 5.7e-15 | 5.3e-12 | 9.4e-12 | 1.4e-11 | 5.5e-11 | 5.1e-10 |

## Заключение

В докладе представлен подход к построению новых методов алгоритмического вычисления производных элементов параметризованных матриц в матричных ортогональных преобразованиях. Результаты работы найдут своё применение в теории адаптивной фильтрации, параметрической идентификации

моделей систем, а также в задачах вычислительной математики, математической физики, оптимизации, управления и др.

## Литература

1. Голуб Дж., Ван Лоун Ч. Матричные вычисления: Пер. с англ. – М.: Мир, 1999.

2. Grewal M.S., Andrews A.P. Kalman Filtering: Theory and Practice Using MATLAB, Second Edition. – John Wiley&Sons, Inc., 2001.

3. Шарый С.П. Курс вычислительных методов. Электронный учебник. – Новосибирск: Новосиб. гос. ун-т., 2014.

4. Kulikova M.V., Tsyganova J.V. Constructing numerically stable Kalman filter-based algorithms for gradient-based adaptive filtering // Int. J. Adapt. Control Signal Process, 2015. (в печати)

5. Семушин И.В. Вычислительные методы алгебры и оценивания: учебное пособие. – Ульяновск: УлГТУ, 2011.