УДК 62-502

В.Б. Ларин

О РЕШЕНИИ ЛИНЕЙНЫХ МАТРИЧНЫХ УРАВНЕНИЙ

Введение

Уравнение Ляпунова (здесь и далее верхний индекс T означает транспонирование)

$$AX + XA^{\mathrm{T}} = C, \tag{1}$$

уравнение Сильвестра

$$X - AXB = C, \tag{2}$$

обобщенное уравнение Сильвестра

$$EX - AXB = C \tag{3}$$

и линейные матричные уравнения более сложной структуры привлекали и продолжают привлекать внимание исследователей (см., например, [1–13] ссылки к ним). Так, в [14] приводятся условия существования решения следующей системы линейных матричных уравнений:

$$\sum_{u=1}^{S_v} A_{uv} X B_{uv} = C_v, \quad v = 1, 2, \ldots, t, \tag{4}$$

где S_v — положительные целые числа. В [15] рассматривается система линейных матричных уравнений:

$$AXB + CYD = M,$$

$$EXF + GYH = N. \tag{5}$$

Все эти уравнения (1)–(5), с использованием аппарата тензорного или Кронекеровского произведения (операция kron.m пакета MATLAB), могут быть сведены к системе линейных алгебраических уравнений (см. п.8.4 [16]). Однако, как отмечено в [2], даже в случае уравнения Ляпунова (1) такой подход не всегда может быть эффективным. Вероятно поэтому разработаны и продолжают разрабатываться алгоритмы решения таких уравнений.

Отметим, что эти уравнения не описывают многих матричных задач механики и теории управления (см., например, [17, 18]). Ниже демонстрируется возможность построения решения уравнений (3)–(5) с использованием аппарата линейных матричных неравенств (ЛМН) [19]. Отметим, что ЛМН применялись в аналогичных задачах (см. [20]). На примерах 1, 2 проводится сравнение предлагаемых вычислительных процедур с известными алгоритмами, в частности с процедурой пакета MATLAB. Рассматривается возможность использования итерационных процедур для повышения точности решения.

1. Общие соотношения

Как отмечено в [19] (соотношения (2.3), (2.4)), матричное неравенство

$$\begin{bmatrix} Q(x) & S(x) \\ S^{\mathrm{T}}(x) & R(x) \end{bmatrix} > 0, \tag{6}$$

где матрицы $Q(x) = Q^{\mathrm{T}}(x)$, $R(x) = R^{\mathrm{T}}(x)$, $S(x)$ линейно зависят от x, эквивалентно следующим матричным неравенствам:

$$R(x) > 0, \;\; Q(x) - S(x)R^{-1}(x)S^{\mathrm{T}}(x) > 0. \tag{7}$$

Рассмотрим следующее ЛМН:

$$\begin{bmatrix} Z & T \\ T^{\mathrm{T}} & I \end{bmatrix} > 0, \;\; Z = Z^{\mathrm{T}}, \;\; Z < \lambda I. \tag{8}$$

Здесь и далее I — единичная матрица соответствующего размера, λ — скаляр. Приняв во внимание (6), (7), соотношения (8) можно переписать следующим образом:

$$Z > T\,T^{\mathrm{T}}, \;\; Z < \lambda I \;\; \text{или} \;\; \lambda I > T\,T^{\mathrm{T}}. \tag{9}$$

Соотношения (9) позволяют рассмотреть следующую стандартную задачу ЛМН на собственные значения (п. 2.2.2 [19]), а именно, задачу минимизации λ при выполнении условий (9).

Соотношения (8) можно обобщить в виде следующей системы ЛМН:

$$\begin{bmatrix} Z & T_i \\ T_i^{\mathrm{T}} & I \end{bmatrix} > 0, \;\; i = 1, 2, \ldots, k \;\; Z = Z^{\mathrm{T}}, \;\; Z < \lambda I, \tag{10}$$

которые можно представить в виде, аналогичном (9):

$$Z > T_{\,i}T_i^{\mathrm{T}}, \;\; i = 1, 2, \ldots, k, \quad Z < \lambda\, I. \tag{11}$$

Применительно к (11) также можно рассматривать стандартную задачу ЛМН на собственные значения и для ее решения использовать процедуру gevp.m пакета MATLAB [21].

2. Обобщенное уравнение Сильвестра

Суть рассматриваемого подхода удобно изложить на примере решения с помощью ЛМН обобщенного уравнения Сильвестра (3). Итак, пусть в (8)

$$T = EX - AXB - C.$$

Используя процедуру gevp.m пакета MATLAB, найдем минимальное значение λ (и соответствующее значение X), при которых выполняются соотношения (9). Очевидно, что при достаточно малой величине λ норма матрицы T также будет достаточно мала, т.е. $T \cong 0$ и, следовательно, полученное в результате использования процедуры gevp.m, значение X можно рассматривать как полученное с определенной точностью решение уравнения (3). В связи с тем, что минимизируется не норма матрицы T, а норма матрицы TT^{T}, можно ожидать снижения точности результата решения (3).

Проиллюстрируем этот подход к решению (3) на примере (6.1) [13].

 ISSN 0572-2691

Пример 1. Итак, фигурирующие в (3) матрицы имеют вид:

$$A=\frac{1}{6}\begin{bmatrix}3&0&0&0\\3&4&1&1\\1&1&3&4\\2&1&0&3\end{bmatrix},\ B=\begin{bmatrix}1&1\\1&1-\varepsilon\end{bmatrix},\ \varepsilon=10^{-2},\ X_0=\begin{bmatrix}1&2\\3&4\\5&6\\7&8\end{bmatrix},$$

$$E=\operatorname{diag}\{1\ 1\ 0\ 1\},\ \ C=EX_0-AX_0B.$$

В результате применения описанной выше процедуры получено решение X, которому соответствует норма погрешности

$$nx=\left\|X-X_0\right\|_\infty=2{,}2\cdot 10^{-3}.$$

При этом процедуру минимизации матрицы $T\,T^{\mathrm{T}}$ можно считать достаточно эффективной. Так, полученному значению X соответствует следующая величина нормы матрицы TT^{T}: $nt=\|TT^{\mathrm{T}}\|_\infty=5{,}5\cdot 10^{-13}$. Естественно, что использование процедур [10, 13], ориентированных на решение этого уравнения, позволяет получить решение с существенно большей точностью (вопросы уточнения полученных решений см. в разд. 5). В этой связи заметим, что в обычных алгоритмах решения этого уравнения трудно учесть особенности искомого решения, в частности находить решение в классе симметричных матриц. В то же время рассматриваемый алгоритм, базирующийся на процедурах ЛМН, позволяет легко учесть такую особенность искомого решения. Другими словами, в ряде задач рассматриваемый алгоритм может дать возможность найти решение (3), в то время как обычные алгоритмы могут не быть эффективными. Проиллюстрируем это на примере.

Пример 2. Пусть матрицы, определяющие уравнение (3), имеют вид:

$$A=\frac{1}{6}\begin{bmatrix}3&0&0&0\\3&4&1&1\\1&1&3&4\\2&1&0&3\end{bmatrix},\ B=\begin{bmatrix}0&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&0&1\end{bmatrix},\ X_0=\begin{bmatrix}1&2&3&4\\2&2&1&5\\3&1&3&6\\4&5&6&4\end{bmatrix},\ E=-\begin{bmatrix}1&2&3&4\\4&5&6&7\\0&0&0&0\\0&0&0&0\end{bmatrix},$$

$$C=EX_0-AX_0B.$$

В связи с тем, что матрица A имеет обратную, (3) можно свести к уравнению Ляпунова

$$A^{-1}C=A^{-1}EX-XB. \tag{12}$$

Если при нахождениях решения (12) использовать стандартную процедуру lyap.m пакета MATLAB, то получим решение, существенно отличающееся от X_0, а именно:

$$nx=\left\|X-X_0\right\|_\infty=15{,}49.$$

Дело в том, что ранг матриц $A^{-1}E$ и B равен 2, т.е. в этом примере нельзя использовать стандартную процедуру lyap.m. Существенно, что при использовании этой процедуры нельзя отыскать решение этого примера в классе симметричных матриц.

Использование в этом примере процедуры, базирующейся на аппарате ЛМН, легко позволяет учесть требование симметричности матрицы искомого решения. Так, использование аппарата ЛМН, позволило получить решение примера, которому соответствуют следящие характеристики погрешностей:

$$nx = \left\| X - X_0 \right\|_\infty = 7{,}7 \cdot 10^{-7}, \;\; nt = \left\| TT^{\mathrm{T}} \right\|_\infty = 7{,}1 \cdot 10^{-30}.$$

Таким образом, в данном примере не удалось получить решение с помощью стандартной процедуры MATLAB, однако алгоритм, базирующийся на ЛМН, оказался вполне «работоспособным».

3. Система уравнений (4)

Очевидно, соотношения (4) можно представить в следующей эквивалентной форме:

$$C_i + \sum_{u=1}^{S_i} A_{ui} X B_{ui} = 0, \;\; i = 1, 2, \ldots, k. \tag{13}$$

Предположим, что одинаковые размеры матриц C_v в (4) и соответствующих матриц C_i в (13) (в ряде случаев это условие может быть нарушено, см. пример 3). Итак, пусть в (10) матрицы T_i имеют вид

$$T_i = C_i + \sum_{u=1}^{S_i} A_{ui} X B_{ui}, \;\; i = 1, 2, \ldots, k. \tag{14}$$

При таких исходных данных, используя применительно к (10) процедуру gevp.m пакета MATLAB, можно найти решение системы (4) или (13). Как уже упоминалось, использование процедур, базирующихся на ЛМН, позволяет накладывать определенные ограничения на матрицу X. Так, можно искать решение (13) в классе симметричных матриц. Проиллюстрируем это на примере решения системы двух уравнений.

Пример 3. Итак, пусть в (14)

$$T_1 = C_1 + A_1 X, \;\; T_2 = C_2 + A_2 X B_2, \;\; A_1 = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \; A_2 = \begin{bmatrix} 7 & 8 & 9 \\ 10 & 11 & 12 \\ 1 & 1 & 2 \end{bmatrix}, \tag{15}$$

$$B_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; X_0 = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 2 & 1 \\ 3 & 1 & 3 \end{bmatrix}, \; C_1 = -A_1 X_0, \;\; C_2 = -A_2 X_0 B_2.$$

Отметим, что при исходных данных (15) ранг матрицы соответствующей линейной системы, определяющей элементы матрицы X, равен 7 (подробности см. п. 8.4 [16]), в то время как размер искомой матрицы равен 3×3, т.е. число неизвестных равно 9 и, следовательно, система не имеет единственного решения. Однако если наложить условие, что искомая матрица является симметричной (т.е. число неизвестных равно 6), то в этом случае система (13) при исходных данных (15) однозначно определяет матрицу X.

Отметим, что в данном примере, матрица C_1 и соответственно T_1 имеют размер 2×3, в то время как матрицы C_2, T_2 имеют размер 3×3.

В этой связи пополним матрицу A_1 нулевой строкой, а именно, будем считать, что матрица A_1 имеет вид

$$A_1 = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 0 & 0 & 0 \end{bmatrix}. \tag{16}$$

 ISSN 0572-2691

С одной стороны, такая замена не изменит условия задачи, с другой стороны, матрицы T_1 и T_2 будут иметь одинаковый размер 3×3.

Итак, в результате решения этого примера, при условии, что разыскивается симметричная матрица X, а матрица A_1 определяется (16), получено решение X, которому соответствует следующая норма погрешности:

$$n = \| X - X_0 \|_\infty = 7{,}7 \cdot 10^{-7}.$$

Этому значению X соответствуют следующие нормы матриц $T_1 T_1^{\mathrm{T}}$, $T_2 T_2^{\mathrm{T}}$:

$$nt_1 = \| T_1 T_1^{\mathrm{T}} \|_\infty = 1{,}9 \cdot 10^{-15},$$

$$nt_2 = \| T_2 T_2^{\mathrm{T}} \|_\infty = 8{,}9 \cdot 10^{-15}.$$

Таким образом, в рассматриваемом примере, описанный выше алгоритм оказался достаточно эффективным.

4. Система уравнений (5)

Как и в случае системы (4), перепишем уравнение (5) в эквивалентной форме, аналогичной (13):

$$M + AXB + CYD = 0, \tag{17}$$

$$N + EXF + GYH = 0.$$

Далее, как и разд. 3, будем предполагать, что размеры матриц M и N совпадают. Итак, пусть в (10) матрицы T_i имеют вид:

$$T_1 = M + AXB + CYD,$$

$$T_2 = N + EXF + GYH.$$

Используя применительно к (10) процедуру gevp.m пакета MATLAB, при таких исходных данных, как и в случае системы (13), можно найти решение системы (17) или (5). Проиллюстрируем это на примере.

Пример 4. Матрицы, фигурирующие в (17), имеют вид:

$$A = -\operatorname{diag}\{1\,1\,0\,1\},\ B = I,$$

$$C = \frac{1}{6}\begin{bmatrix} 3 & 0 & 0 & 0 \\ 3 & 4 & 1 & 1 \\ 1 & 1 & 3 & 4 \\ 2 & 1 & 0 & 3 \end{bmatrix},\ D = \begin{bmatrix} 1 & 1 \\ 1 & 1-\varepsilon \end{bmatrix},\ \varepsilon = 10^{-2},$$

$$E = C, F = D, G = 2C^{\mathrm{T}}, H = 3D,$$

$$X_0 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix},\ Y_0 = \begin{bmatrix} 8 & 7 \\ 6 & 5 \\ 4 & 3 \\ 2 & 1 \end{bmatrix},$$

$$M = -AX_0B - CY_0D,$$

$$N = -EX_0F - GY_0H.$$

При таких исходных данных, используя процедуру gevp.m пакета MATLAB, получены значения X и Y, которым соответствуют нормы погрешностей

$$nx = \| X - X_0 \|_\infty = 4{,}8 \cdot 10^{-3}, \quad ny = \| Y - Y_0 \|_\infty = 5{,}7 \cdot 10^{-4}.$$

Нормы величин невязок следующие:

$$nt_1 = \| T_1 T_1^{\mathrm{T}} \|_\infty = 1{,}9 \cdot 10^{-28}, \quad nt_2 = \| T_2 T_2^{\mathrm{T}} \|_\infty = 2{,}36 \cdot 10^{-11}.$$

Таким образом, можно констатировать эффективность использования предложенного алгоритма при решении данного примера.

5. Уточнение полученных решений

Выше отмечалось, что рассматриваемый подход может уступать в точности результата алгоритмам, ориентированным только на решение того или иного типа уравнений. Имеется возможность с помощью итерационных процедур повысить точность рассматриваемых алгоритмов. В частности, можно использовать итерационные процедуры, аналогичные тем, которые в [22] применяются для построения решения уравнения (4) при $v = 1$. Существенно, что в этих итерационных процедурах могут использоваться описанные выше алгоритмы.

Итак, рассмотрим уравнение (4) при $v = 1$. Пусть известно начальное приближение решения этого уравнения, которое обозначим X_1. Аналогичная [22] итерационная процедура уточнения полученного решения выглядит следующим образом. Будем искать уточненное решение X_2 в виде

$$X_2 = X_1 + \varepsilon X_1, \tag{18}$$

где εX_1 — искомая поправка. Имеем (в обозначениях (4) $v = 1$, $S_v = k$):

$$\sum_{u=1}^{k} A_u \varepsilon X_1 B_u = C_1, \tag{19}$$

$$C_1 = C - \sum_{u=1}^{k} A_u X_1 B_u.$$

Отметим, что уравнение (19), определяющее поправку εX_1, имеет структуру (4) и, следовательно, для итерационной процедуры уточнения решения (4) (нахождения εX_1) могут использоваться алгоритмы, описанные в разд. 2, 3.

Аналогичный подход может применяться и для уточнения решения системы (5). Так, имея начальное приближение решения (5), которое обозначим X_1, Y_1, уточненное значение этого решения будем искать в виде

$$X_2 = X_1 + \varepsilon X_1, \; Y_2 = Y_1 + \varepsilon Y_1. \tag{20}$$

Подставив (20) в (5), будем иметь:

$$A\varepsilon X_1 B + C\varepsilon Y_1 D = M_1, \; M_1 = M - AX_1B + CY_1D,$$

$$E\varepsilon X_1 F + G\varepsilon Y_1 H = N_1, \; N_1 = N - EX_1F - GY_1H.$$

Таким образом, для определения εX_1, εY_1 можно использовать процедуры, описанные в разд. 4.

Проиллюстрируем возможность повышения точности на рассмотренных выше примерах. Итак, в примере 1 после двух описанных выше итерационных процедур получено значение X, решения (3), которому соответствуют

$$nx = \|X - X_0\|_\infty = 3{,}48 \cdot 10^{-10}.$$

Сравнивая эту норму погрешности с результатом, полученном в примере 1, можно констатировать уменьшение нормы величины погрешности на семь порядков.

Рассмотрим пример 4. В результате уточнения решения примера 4 при использовании двух описанных выше итерационных процедур получены значения X, Y, решения (5), погрешности которых следующие:

$$nx = \|X - X_0\|_\infty = 3{,}62 \cdot 10^{-4},\ ny = \|Y - Y_0\|_\infty = 4{,}4 \cdot 10^{-5}.$$

Сравнивая полученные значения n_x, n_y, с приведенными в примере 4, можно констатировать повышение точности на порядок. Однако в этом примере дальнейшее увеличение числа итераций не сопровождается повышением точности результата.

Заключение

Предложен алгоритм нахождения решений различных типов линейных матричных уравнений. Алгоритм базируется на аппарате линейных матричных неравенств. При использовании данного алгоритма отмечается возможность, принимать во внимание особенности разыскиваемого решения (симметричная матрица и т.п.). На примерах решения различных типов линейных матричных уравнений демонстрируется эффективность предлагаемого алгоритма.

В.Б. Ларін

ПРО РОЗВ'ЯЗОК ЛІНІЙНИХ МАТРИЧНИХ РІВНЯНЬ

Для знаходження розв'язків різних типів лінійних матричних рівнянь запропоновано алгоритм, що базується на апараті лінійних матричних нерівностей. Зазначається, що при використанні даного алгоритму слід враховувати особливості шуканого рішення (симетрична матриця і т.п.). Розглянуто можливість використання ітераційних процедур для підвищення точності розв'язків. На прикладах розв'язків різних типів лінійних матричних рівнянь показано ефективність запропонованого алгоритму.

V.B. Larin

ON SOLUTION OF THE LINEAR MATRIX EQUATIONS

The algorithm of finding solutions of various types of the linear matrix equations is offered. The algorithm is based on the linear matrix inequalities. It is noted the opportunity, to take into account the feature of the searched solution (a symmetric matrix, etc.) at use of the given algorithm. On the examples of solving various types of the linear matrix equations, the efficiency of offered algorithm is shown.

1. *Jones J.* Solution of certain matrix equations // Proc. American Math. Society. — 1972. — **31**. — P. 333–339.
2. *Pace I.S., Barnett S.* Comparison of numerical methods for solving Lyapunov matrix equations // Int. J. Control. — 1972. — **15**. — P. 907–915.
3. *Шестопал В.Е.* Решение матричного уравнения $AX - XB = C$ // Математические заметки. — 1975. — **19**, № 3. — С. 449–451.
4. *Ларин В.Б.* Решение матричного уравнения // Математическая физика. — 1977. — Вып. 22. — С. 12–14.
5. *Gardiner J.D., Laub A.J., Amato J.J., Moler C.B.* Solution of the Sylvester matrix equation $AXB^{\mathrm{T}} + CXD^{\mathrm{T}} = E$ // ACM Transactions on Math. Software. — 1992. — **18**, N 2. — P. 223–231.
6. *Aliev F.A., Larin V.B.* Optimization of linear control systems: analytical methods and computational algorithms. — Amsterdam: Gordon and Breach Science Publishers, 1998. — 261 p.
7. *Jiang T., Wei M.* On solutions of the matrix equations $X - AXB = C$ and $X - A\overline{X}B = C$ // Linear Algebra and its Applications. — 2003. — **367**. — P. 225–233.
8. *Adam M., Assimakis N., Sanida F.* Algebraic solutions of the matrix equations // Intern. Journal of Algebra. — 2008. — **2**, N 11. — P. 501–518.
9. *Wu A.G., Duan G.R., Zhou B.* Solution to generalized Sylvester matrix equations // IEEE Trans. Automat. Control. — 2008. — **53**, N 3. — P. 811–815.
10. *Ларин В.Б., Алиев Ф.А.* Об использовании соотношения Баса при решении матричных уравнений // Докл. НАН Азербайджана. — 2008. — № 4. — С. 15–25.
11. *Larin V.B.* On solutions of the Lyapunov equations // Appl. and Computational Math. — 2008. — **7**, N 2. — P. 162–167.
12. *Ларин В.Б.* О решении уравнения Сильвестра // Проблемы управления и информатики. — 2009. — № 1. — С. 29–34.
13. *Larin V.B.* Solution of matrix equations in problems of the mechanics and control // Int. Appl. Mech. — 2009. — **45**, N 8. — P. 847–872.
14. *Zhang X.* A Remark on common solutions of a pair of matrix equation // Acta Math. Univ. Comenianae — 2004. — **LXXII**, N 2. — P. 151–154.
15. *Dehghan M., Hajarian M.* An iterative method for solving the coupled Sylvester matrix equations over generalized bisymmetric matrices // Appl. Math. Modelling — 2010. — **34.** — P. 639–654.
16. *Ланкастер П.* Теория матриц. — М.: Наука, 1978. — 280 с.
17. *Lukyanova T.A., Martynyuk A.A.* Sufficient conditions of connective stability of motion on time scale // Int. Appl. Mech. — 2013. — **49**, N 2. — P. 232–244.
18. *Larin V.B., Tunik A.A.* On inertial-navigation system without angular-rate sensors // Ibid. — 2013. — **49**, N 4. — P. 488–500.
19. *Boyd S., Ghaoui L.E., Feron E., Balakrishnan V.* Linear matrix inequalities in system and control theory. — Philadelphia: SIAM, 1994. — 193 p.
20. *Larin V.B.* Algorithms for solving a unilateral quadratic matrix equations and model updating problem // Int. Appl. Mech. — 2014. — **50**, N 3. — P. 321–334.
21. *Gahinet P., Nemirovski A., Laub A.J., Chilali M.* LMI control toolbox users guide. — The MathWorks Inc. — 1995. — P. 130.
22. *Kressner D.*, *Sirkovic P.,* Truncated low-rank methods for solving general linear matrix equations // Numer. Linear Algebra Appl. — 2015. — **22.** — P. 564–583.

Получено 03.06.2015