УДК 519.6 DOI: https://doi.org/10.20998/2078-9130.2020.1.221713

В.М. ГРИЩЕНКО

ЕФЕКТИВНІСТЬ ЕЛЕМЕНТАРНИХ НЕУНІТАРНИХ ПЕРЕТВОРЕНЬ В ПОЄДНАННІ З ВІРТУАЛЬНИМИ МАТРИЦЯМИ В ОРГАНІЗАЦІЇ ОБЧИСЛЕНЬ МАТРИЧНОЇ АЛГЕБРИ

Сучасні прикладні задачі матричної алгебри потребують алгоритмів ефективної та надійної роботи з великою кількістю рівнянь. Ціль таких алгоритмів полягає в перетворенні цих рівнянь з широким діапазоном чисельних значень їх коефіцієнтів. Перетворення спрямовані на спрощення форми матриць з метою надання їм однієї з канонічних. Розроблено багато чисельних методів, схем, алгоритмів для досягнення цих цілей. На якість результатів перетворень безумовно впливає не лише сам метод а також і порядок виконання окремих операцій. В даній роботі запропоновано прийнятний (на наш погляд) варіант роботи з відомими методами та схему їх застосування коли пріоритетні для обчислень питання точності та стійкості, економії оперативної пам'яті та раціональної кількості операцій узгоджуються між собою. Базовою в цих перетвореннях прийнята елементарна неунітарна матриця. Для обмеження безконтрольного росту коефіцієнтів призначено певний порядок операцій з використанням Virtual матриць, в основі яких управління перестановками рядків / стовбців. Ланцюги перетворень з використанням елементарних матриць формують циклічні процеси та можуть бути універсальними. В роботі наведені приклади використання запропонованих схем для двох важливих типових матричних перетворень.

Ключові слова: елементарна неунітарна матриця; Virtual, Original матриці; Pointer рядків/стовбців; схема алгоритму.

В.Н. ГРИЩЕНКО

ЭФФЕКТИВНОСТЬ ЭЛЕМЕНТАРНЫХ НЕУНИТАРНЫХ ПРЕОБРАЗОВАНИЙ СОВМЕСТНО С ВИРТУАЛЬНЫМИ МАТРИЦАМИ В ОРГАНИЗАЦИИ ВЫЧИСЛЕНИЙ МАТРИЧНОЙ АЛГЕБРЫ

Современные прикладные задачи матричной алгебры нуждаются в алгоритмах эффективной и надежной работы с большим количеством уравнений. Цель таких алгоритмов состоит в преобразовании этих уравнений с широким диапазоном численных значений их коэффициентов. Преобразования направлены на упрощение формы матриц с целью придания им одной из канонических. Разработано много численных методов, схем, алгоритмов для достижения этих целей. На качество результатов преобразований безусловно влияет не только сам метод но также и порядок выполнения отдельных операций. В данной работе предложен приемлемый (на наш взгляд) вариант работы с известными методами и схему их использования когда приоритетные для вычислений вопросы точности и устойчивости, экономии оперативной памяти и рационального количества операций согласуются между собою. Базовой в этих преобразованиях принята элементарная неунитарная матрица. Для ограничения бесконтрольного роста коэффициентов назначен определенный порядок операций с использованием Virtual матриц, в основе которых управление перестановками рядков / столбцов. Цепочки преобразований с использованием элементарных матриц формируют циклические процессы и могут быть универсальными. В работе приведены примеры использования предложенных схем для двух важных типовых матричных преобразований.

Ключевые слова: элементарная неунитарная матрица; Virtual, Original матрицы; Pointer рядков/столбцов; схема алгоритма.

V.M. GRISCHENKO

EFFICIENCY OF ELEMENTARY NON-UNITARY TRANSFORMATIONS TOGETHER WITH VIRTUAL MATRICES IN THE ORGANIZATION OF CALCULATIONS OF MATRIX ALGEBRA

Modern applied problems of matrix algebra require algorithms for efficient and reliable operation with a large number of equations. The purpose of such algorithms is to transform these equations with a wide range of numerical values of their coefficients. The transformations are aimed at simplifying the form of matrices in order to make them one of the canonical ones. Many numerical methods, schemes, algorithms have been developed to achieve these goals. The quality of the transformation results is certainly influenced not only by the method itself, but also by the order in which individual operations are performed. In this paper, an acceptable (in our opinion) version of working with known methods and a scheme for their use is proposed when the issues of accuracy and stability, saving RAM and a rational number of operations are consistent with each other. An elementary non-unitary matrix is accepted as the basic one in these transformations.

To limit the uncontrolled growth of the coefficients, a certain order of operations is assigned using Virtual matrices, which are based on the management of row / column permutations. Chains of transformations using elementary matrices form cyclical processes and can be universal. The paper gives examples of using the proposed schemes for two important typical matrix transformations.

Key words: elementary non-unitary matrix; Virtual, Original matrices; Pointer of rows / columns; algorithm diagram.

1. Вступ . Необхідність рішення сучасних прикладних задач, в яких виникає потреба чисельних операцій з матричними рівняннями, потребує особливої уваги до ефективності розрахункових методів та схем організації обчислень. Запропоновано багато алгоритмів для різноманітних задач лінійної алгебри, які базуються на значному арсеналі як теоретичних так і практичних підходів. Вони детально розроблені, робота їх продемонстрована на чисельних прикладах та викладена в багаточисельній як науковій літературі так і в підручниках [1-13].

$$k\cdot x = b;$$

$$k\cdot x = \lambda\cdot m\cdot x; \qquad \det(k - \lambda\cdot m) = 0.$$

Для задач невеликого розміру використання тих чи інших розрахункових схем може не мати принципового значення. Але для рівнянь значного порядку, безумовно, орієнтованих на комп'ютерне моделювання, вимоги до якості чисельних методів та схем застосування стають більш жорсткими. Так, не викликає заперечень, що важливою складовою схем рішення таких матричних рівнянь є використання ланцюгів елементарних матриць для цілеспрямованого перетворення їх загальних форм до простих. В основі роботи елементарних перетворень лежить циклічне вилучення ненульових позицій [1, 3, 4, 6].

Чисельні схеми, які працюють в потужних комп'ютерних системах, повинні задовольняти ряду вимог: забезпеченню точності результатів та стійкості роботи алгоритму, економічності використання оперативної пам'яті РС та раціональної кількості операцій, тощо. У цьому зв'язку є зауваження по використанню однієї з найбільш відомих схем перетворення матричних рівнянь. В практиці інженерних розрахунків виключно широкого розповсюдження отримало використання елементарних ортогональних матриць. В деяких випадках матриця обертань стала опорною в операціях таких алгоритмів. Але можна зауважити, що її використання не задовольняє деяким з наведених вище вимог. Поряд з важливими перевагами вона вносить суттєвий недолік, який стає відчутним при великій кількості операцій. Так, один рядок обертань виконує «корисну» функцію, а другий – є «паразитним», так що загальна кількість операцій зростає майже вдвічі. Для виправлення цього «недоліку» виникає пропозиція вилучити з розрахункового алгоритму матрицю обертань, а натомість замінити її елементарною неунітарною матрицею, яка виконує лише «корисну» функцію – вилучення з рівнянь однієї з ненульових позицій. Через таку заміну в обчислювальному процесі можуть виникати серйозні проблеми, які потрібно уже компенсувати схемою роботи алгоритму.

2. Постановка задачі. Таким чином, аналіз показав, що сучасні інженерні задачі матричної алгебри потребують алгоритмів роботи з великою кількістю рівнянь. Їх задача полягає в перетворенні цих рівнянь з широким діапазоном чисельних значень елементів. Стратегія цих перетворень пов'язана зі спрощенням форм матриць – переходу до однієї з канонічних. Засобами для досягнення цих цілей є чисельні методи, яких розроблено немало.

Задача цієї роботи – розглянути найбільш прийнятний (раціональний) варіант методу та схему його застосування, в якому узгоджуються пріоритетні питання точності, стійкості роботи, економії оперативної пам'яті, мінімальної кількості операцій тощо. Важливим також є фактор універсальності підходу.

Зважаючи на це вилучимо з розгляду елементарне унітарне перетворення типу обертань, а натомість основною для використання в схемі обчислень приймемо елементарну неунітарну матрицю S_{ij} (1):

$$S_{ij} = \begin{array}{c} \\ \\ \\ i \\ \\ \\ \end{array}\begin{array}{c}\begin{bmatrix} 1 & . & . & . & . \\ . & 1 & . & . & . \\ . & . & 1 & . & . \\ . & . & s_{ij} & 1 & . \\ . & . & . & . & 1 \end{bmatrix} \\ j \end{array}; \quad S_{ij}^{-1} = \begin{array}{c} \\ \\ \\ i \\ \\ \\ \end{array}\begin{array}{c}\begin{bmatrix} 1 & . & . & . & . \\ . & 1 & . & . & . \\ . & . & 1 & . & . \\ . & . & -s_{ij} & 1 & . \\ . & . & . & . & 1 \end{bmatrix} \\ j \end{array}; \quad (1)$$

Важливою деталлю є те, що ця матриця допускає просте обчислення зворотної (1). Вибравши цей варіант потрібно прийняти до уваги, що при безконтрольному ході алгоритму чисельні значення величин sij можуть стати необмеженими, що приведе до втрати стійкості алгоритму. Тому для стабілізації обчислювального процесу в таких випадках вводиться наступне обмеження :

$$|\, s_{ij} \,| \le 1. \qquad (2)$$

Виконання цієї умови можна забезпечити лише шляхом організації такої схеми обчислень, коли виконуються перестановки елементів рядків (стовбців) матриць.

В свою чергу такий сценарій є абсолютно неприйнятний в силу катастрофічного росту числа операцій процесу. Для виходу з останньої проблеми в роботі пропонується використовувати вказівники рядків та стовбців (Pointer – вектори). Це дозволяє міняти місцями не самі елементи матриць а лише їх номери в Pointer.

Таким чином, формулюється задача побудови такої схеми перетворення матричних рівнянь (яка в свою чергу може бути складовою різних обчислювальних процесів), при якій:

- використовуються ланцюги лише з елементарних неунітарних матриць типу S_{ij};
- чисельні значення цих матриць s_{ij} обмежені величиною: $|\, s_{ij} \,| \le 1$;
- за допомогою векторів – вказівників (Pointer) замість перестановок елементів матриць здійснюються перестановки самих номерів рядків/стовбців;
- для зберігання елементів активної матриці в оперативній пам'яті РС виділяється лише один масив (Original матриця). Натомість для управління та спостереженням за ходом процесу приведення цієї матриці до канонічної форми вводиться умовна матриця (Virtual – матриця). Вона формується лише на папері і її елементи можуть бути визначені по Original формі. В результаті обчислювального процесу згідно алгоритму до канонічного виду буде приведена віртуальна форма матриць.

3. Основні положення алгоритму. Для реалізації основних положень алгоритму введені деякі терміни та визначення.

- $i,j = 1, 2, 3, \ldots n;$ − порядкові номери рядків / стовбців матриці;
- $m_0[i, j]$ – *оригінальна* (Original) форма матриці m. Це реальна звичайна форма активної матри-

ці, елементи якої та всі зміни в них зберігаються у виділеній оперативній пам'яті РС;

- $I(i)$, $J(j)$ – *вектори-вказівники* (Pointer) номерів рядків та стовбців матриці. На старті обчислювального процесу їм призначаються номери елементів натурального ряду:
  $I(i) = \{1, 2, 3, \ldots n\}$; $J(j) = \{1, 2, 3, \ldots n\}$.
  В ході алгоритму в разі необхідності виконуються перестановки цих номерів ;
- $m_V[i, j]$ – *віртуальна* (Virtual) форма матриці m. Це форма умовної матриці, на базі якої імітується процес приведення її до каноніки. Ця форма записана лише на папері, а значення елементів можна відтворити по відповідній Original так:

$$m_V[i, j] = m_0[I(i), J(j)]. \quad (3)$$

Суть алгоритму полягає в тому, що на базі Virtual матриці планується та реалізується процес приведення її до канонічного виду. Для контролю за виконанням обмежень $|s_{ij}| \le 1$ в разі необхідності здійснюються перестановки номерів рядків/стовбців у відповідних Pointer. Фізично ж чисельні значення елементів завжди зберігаються в її Original формі без здійснення таких перестановок. При такій організації процесу вдається утримати баланс між пріоритетними вимогами до обчислювального процесу, зберігаючи при цьому контроль за величинами основних проміжних обчислень алгоритму. Натомість добре відома думка, що суттєвим джерелом нестійкості в процесах перетворень може бути значне зростання величин проміжних обчислень алгоритму. Так ось. єдина можливість в певних межах регулювати зростання елементів проміжних обчислень – це використання перестановок при реалізації процесу, які імплементовані в алгоритмі. Тобто ці питання пов'язані з точністю, стійкістю при економії числа операцій та ОП РС.

Реалізація алгоритму відбувається у формі побудови комбінацій правих та лівих циклічних перетворень, які генерують ланцюги, що складаються лише з елементарних неунітарних матриць, та мають форму подібну такій:

$$S_{\Sigma} = S_{21} \cdot S_{32} \cdot S_{43} \cdots S_{ij} \,. \quad (4)$$

- Transformer − цим терміном будемо вирізняти матриці-перетворення типу (4). Ці матриці мають особливі якості. В кожній з них, що входить в Transformer, величина елементів обмежена одиницею; вони добре обумовлені, визначники як кожної так і всього перетворення дорівнюють одиниці. Ну і, як наслідок, для перетворення типу Transformer легко знаходиться зворотне перетворення, яке також належить цьому типу.

Таким чином, сформульована схема організації перетворень в матричній алгебрі, яка має свої переваги. Особливості роботи цього підходу розглянуті на прикладі двох важливих в чисельних методах лінійної алгебри задачах. Це розкладання матриці на множники, та одночасне приведення пучка двох матриць до форми Хесенберга.

4. Розкладання матриці на множники. Важливу роль в чисельних методах лінійної алгебри відіграє розкладання матриці на співмножники, наприклад, *QR*-розкладання. В принципі кожне перетворення є таким розкладанням. Відомо існування таких еквівалентних перетворень, при яких матриця приводиться до трикутної, Хесенберга, Жордана та інших. Існує думка, що більш ефективними є ті розкладання на множники, які пов'язані з використанням послідовних виключень елементів за допомогою елементарних матриць. В даній роботі розглядаються виключення з допомогою Transformer-перетворень. Такі розкладання вигідні, зокрема тим, що обумовленість матриці при цьому не змінюється, так як визначник Transformer дорівнює одиниці.

Т е о р е м а 1. *Довільна квадратна матриця m за допомогою правої Transformer-матриці u може бути приведена до Original-матриці* $m_0 = m \cdot u$, *віртуальна для якої* $m_V[i, j]$ *є правою трикутною, де*

$$m_V[i,j] = m_0[i, J(j)].$$

Доказ цієї теореми є конструктивним втіленням запропонованої схеми. Розглядається приведення (розкладання) довільної квадратної матриці m до правої трикутної за допомогою ланцюга правих елементарних неунітарних матриць типу S_{ij} (Transformer) за схемою: $m{\cdot}u \;{-}{>}\; R$.

Основні кроки схеми детально розглядаються на прикладі матриці (5х5) з набором простих чисел:

$$m[i, j] = \begin{bmatrix} 5 & 1 & . & . & . \\ 5 & 1 & 1 & . & . \\ 5 & 2 & 1 & 1 & . \\ 5 & 3 & 2 & 1 & 1 \\ 5 & 4 & 3 & 2 & 1 \end{bmatrix}; \quad (5)$$

а) На старті формуються початкові значення основних елементів процесу:

-) $J(j)=\{\ 1,\ 2,\ 3,\ 4,\ 5\ \}$− вектор Pointer номерів стовбців матриці;
-) $m_0 = m_V = m$ – Original та Virtual форми матриці m, які співпадають зі стартовою;
-) $u= E$ − стартова матриця Transformer – правих перетворень до каноніки.

Приведення представляє собою циклічний процес вилучення ненульових позицій в $m_V[i,\ j]$ правими перетвореннями в такому порядку :

– останнього рядка*:*

[5, 1], [5, 2], [5, 3], [5, 4];

– потім попередніх рядків:

[4, 1], [4, 2], [4, 3];
[3, 1], [3, 2];
[2 , 1].

б) На першому кроці вилучається позиція [5, 1] з допомогою перетворення S_{21}.

Можна сказати, що використовується стратегія вибору ведучого елемента з двох сусідніх позицій 5-го рядка : [5, 1] і [5, 2]. З метою врегулювання величини s_{21} змінюється конфігурація вектора Pointer J, що відповідає перестановці 1-го та 2-го стовбців:

$$J = \{2, 1, 3, 4, 5\}; \quad s_{21} = -4/5.$$

Original та Virtual форми змінюються за формулами правих перетворень:

*) $m_0[i,j] \cdot S_{J(2),J(1)} \text{–>} m_0[i,j]'$,

$$\begin{bmatrix} 5 & 1 & . & . & . \\ 5 & 1 & 1 & . & . \\ 5 & 2 & 1 & 1 & . \\ 5 & 3 & 2 & 1 & 1 \\ 5 & 4 & 3 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & -4/5 & . & . & . \\ . & 1 & . & . & . \\ . & . & 1 & . & . \\ . & . & . & 1 & . \\ . & . & . & . & 1 \end{bmatrix} \to \begin{bmatrix} 5 & -3 & . & . & . \\ 5 & -3 & 1 & . & . \\ 5 & -2 & 1 & 1 & . \\ 5 & -1 & 2 & 1 & 1 \\ 5 & \bullet & 3 & 2 & 1 \end{bmatrix};$$

$$\text{*) } m_V[i,\ ] = m_0[i, J(j)] = \begin{bmatrix} -3 & 5 & . & . & . \\ -3 & 5 & 1 & . & . \\ -2 & 5 & 1 & 1 & . \\ -1 & 5 & 2 & 1 & 1 \\ \bullet & 5 & 3 & 2 & 1 \end{bmatrix}.$$

Нагадаємо, що саме у Virtual-формі m_V імітується процес приведення до каноніки. А фізично в пам'яті PC зберігається m_0 (Original-форма), що не буде канонічною.

В матриці u формується праве перетворення-Transformer:

*) $u \cdot S_{J(2)J(1)} \text{–>} u'$.

На цьому закінчується перший крок.

в) На другому і наступних кроках дотримуємось умов циклічності. Вилучається позиція [5, 2] з допомогою перетворення S_{32}. Проводиться вибір ведучого елемента з сусідніх позицій 5-го рядка: [5,2] і [5,3]. З метою врегулювання величини $\boldsymbol{s_{32}}$ знову змінюється конфігурація вектора Pointer, що відповідає перестановці 2-го та 3-го стовбців:

$$J = \{2, 3, 1, 4, 5\}; \quad s_{32} = -3/5.$$

Відповідних змін зазнають Original форма та Transformer. Virtual-матриця може бути відтворена за формулою (3). Очевидно, що при реалізації кожного кроку зберігаються всі нульові позиції, що одержані на попередніх.

Приведемо проміжні результати перетворення матриці $\boldsymbol{m}$ після вилучення ненульових позицій 5-го рядка.

$$J = \{2, 3, 4, 5, 1\}.$$

Original- та Virtual-матриці набувають такого вигляду:

$$m_0 = \begin{bmatrix} 5 & . & -3 & -2 & -1 \\ 5 & -1 & -2 & -2 & -1 \\ 5 & . & -2 & -1 & -1 \\ 5 & . & -1 & -1 & . \\ 5 & \bullet & \bullet & \bullet & \bullet \end{bmatrix};$$

$$m_V = \begin{bmatrix} . & -3 & -2 & -1 & 5 \\ -1 & -2 & -2 & -1 & 5 \\ . & -2 & -1 & -1 & 5 \\ . & -1 & -1 & . & 5. \\ \bullet & \bullet & \bullet & \bullet & 5 \end{bmatrix}.$$

г) Після завершального кроку циклічного процесу буде одержана вся необхідна інформація приведення матриці до правої трикутної форми. Проведемо аналіз одержаних результатів.

$J = \{2, 3, 5, 4, 1\}$ – Pointer-вектор;

$$m_0 = \begin{bmatrix} 5 & \bullet & . & -2 & -1 \\ 5 & \bullet & 1 & -2 & -1 \\ 5 & \bullet & \bullet & -1 & -1 \\ 5 & \bullet & \bullet & -1 & \bullet \\ 5 & \bullet & \bullet & \bullet & \bullet \end{bmatrix} \text{ – Original форма;}$$

$$u_0 = \begin{bmatrix} 1 & -1/5 & . & -2/5 & -1/5 \\ . & 1 & . & . & . \\ . & . & 1 & . & . \\ . & -1 & -1 & 1 & . \\ . & -1 & -1 & . & 1 \end{bmatrix} \text{ – Transformer матриця.}$$

Для перевірки правильності проведених обчислень знайдемо приведення стартової матриці $\boldsymbol{m}\cdot$за допомогою фінального Transformer:

*) $m \cdot u_0$ –>

–>

$$\begin{bmatrix} 5 & 1 & . & . & . \\ 5 & 1 & 1 & . & . \\ 5 & 2 & 1 & 1 & . \\ 5 & 3 & 2 & 1 & 1 \\ 5 & 4 & 3 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & -1/5 & . & -2/5 & -1/5 \\ . & 1 & . & . & . \\ . & . & 1 & . & . \\ . & -1 & -1 & 1 & . \\ . & -1 & -1 & . & 1 \end{bmatrix} \text{–>} m_0.$$

Результати співпадають. Це підтвердження правильності попередніх обчислень.

*) Наостанок, канонічну форму (праву трикутну) можна відтворити згідно (3). Для цього потрібно встановити порядок стовбців m_V як вказано у векторі-Pointer ($J(j)$):

$$m_V[i,j] = m_0[i, J(j)] = \begin{bmatrix} . & . & -1 & -2 & 5 \\ \bullet & 1 & -1 & -2 & 5 \\ \bullet & \bullet & -1 & -1 & 5 \\ \bullet & \bullet & \bullet & -1 & 5 \\ \bullet & \bullet & \bullet & \bullet & 5 \end{bmatrix}. \qquad (6)$$

Звернемо увагу на матрицю-Transformer u_0 –це не випадково, що всі її елементи обмежені, в даному випадку не перевищують 1.

Таким чином, не зменшуючи загальності виконане потрібне перетворення заданої матриці (5) в канонічну (6), а також доказ цієї теореми.

5. Одночасне приведення пучка двох матриць до узагальненої форми Хесенберга. До числа важливих чисельних задач лінійної алгебри відноситься також одночасне двостороннє перетворення кожної з двох матриць пучка. Це, наприклад, операції, що передують задачі обчислення спектра власних значень і власних векторів. В таких задачах щоб пришвидшити збігання ітераційних процесів в пошуку EigenValue попередньо приводять пучок до простого виду..

В даній роботі використовуються виключно Transformer-перетворення. Одне ліве – $v^{\mathrm{T}}[i,j]$, друге праве – $u[i, j]$.

Т е о р е м а 2. *Довільний пучок двох квадратних матриць (k, m) за допомогою лівої v^T та правої* $\boldsymbol{u}$ *Transformer-матриць може бути приведений до пучка двох Original-матриць* $(k_0, m_0) = (v^{\mathrm{T}}ku,\ v^{\mathrm{T}}mu)$, *віртуаль-*

на форма яких (k_V, m_V) буде складатись з форми Хесенберга для $k_V[i,j] = k_0[I(i),J(j)]$ та правої трикутної для $m_V[i,j] = m_0[I(i),J(j)]$.

Доказ цієї теореми також проводиться конструктивно втіленням запропонованої схеми. Є відмінності, які розглянемо більш детально. Розглядається приведення пучка двох довільних квадратних матриць (k, m) до узагальненої форми Хесенберга (правої майже трикутної для k та правої трикутної для m). Перетворення виконуються за допомогою лівого оператора v^T та правого u, кожен з яких складається з елементарних неунітарних матриць типу S_{ij} (Transformer-ланцюги) за схемою: (v^Tku, v^Tmu).

Для перетворень використовується відома в чисельних методах схема:

- *попередньо матриця m приводиться до трикутної форми;*
- *далі розглядається циклічний процес, в якому почергово з допомогою лівих перетворень матрицю k наближають до правої майже трикутної. Ця операція супроводжується пошкодженням трикутної форми матриці m;*
- *тому після кожного лівого з допомогою правих перетворень організується відновлення трикутної форми матриці m.*

Основні кроки схеми детально розглядаються на прикладі пучка 2-х матриць (5x5) з набором простих чисел. При цьому будемо вважати, що на старті матриця ***m*** вже приведена до трикутної форми, як показано в першому прикладі, так що пучок має вигляд:

$$(k,m)=\left(\begin{bmatrix}1&-1&.&.&.\\2&1&-1&.&1\\3&2&1&5&.\\4&3&2&.&.\\5&5&.&.&.\end{bmatrix},\ \begin{bmatrix}1&1&1&.&.\\.&1&1&.&.\\.&.&1&5&1\\.&.&.&5&3\\.&.&.&.&5\end{bmatrix}\right);\quad (7)$$

а) На початку формуються стартові значення основних елементів процесу:

-) $I(i)=\{1, 2, 3, 4, 5\}$; $J(j)=\{1, 2, 3, 4, 5\}$ − вектори Pointer номерів рядків/стовбців матриць;
  $k_V = k_0= k$;
-) $m_V = m_0 = m$; –Virtual та Original форми матриць, які співпадають зі стартовими;
- $v^T = u = E$ − стартові Transformer-матриці лівих та правих перетворень.

Циклічний процес вилучення ненульових позицій в $k_V[i, j]$ за допомогою лівого Transformer (v^T) відбувається в такому порядку :

– елементи першого стовбця*:*

[5, 1], [4, 1], [3, 1];

– потім другого і третього:

[5, 2], [4, 2];

[5, 3].

При цьому кожне ліве перетворення супроводжується пошкодженням трикутної форми матриці m_V в таких позиціях:

– [5, 4], [4, 3], [3, 2];
– [5, 4], [4, 3];
– [5, 4].

Як було вказано вище після кожного лівого виконується відповідний правий Transformer–u . Задача останнього відновити завдану шкоду m_V при збереженні всіх нульових позицій, які були одержані на попередніх кроках.

б) На першому кроці вилучається позиція [5, 1] матриці k_V з допомогою лівого перетворення S_{54}.

Можна сказати, що використовується стратегія вибору ведучого елемента з двох сусідніх позицій 1-го стовбця: [5, 1] і [4, 1]. З метою врегулювання величини s_{54} змінюється конфігурація вектора Pointer I, що відповідає перестановці 4-го та 5-го рядків матриць:

$I =\{1, 2, 3, 5, 4\}$; s_{54} = -4/5.

Original та Virtual форми пучка змінюються за формулами лівого перетворення:

*) $S_{I(5),I(4)}\cdot(k_0, m_0)$ –>

$$\to\begin{bmatrix}1&.&.&.&.\\.&1&.&.&.\\.&.&1&.&.\\.&.&.&1&-4/5\\.&.&.&.&1\end{bmatrix}\cdot\left(\begin{bmatrix}1&-1&.&.&.\\2&1&-1&.&1\\3&2&1&5&.\\4&3&2&.&.\\5&5&.&.&.\end{bmatrix},\begin{bmatrix}1&1&1&.&.\\.&1&1&.&.\\.&.&1&5&1\\.&.&.&5&3\\.&.&.&.&5\end{bmatrix}\right)\to$$

$$\to\cdot\left(\begin{bmatrix}1&-1&.&.&.\\2&1&-1&.&1\\3&2&1&5&.\\\bullet&-1&2&.&.\\5&5&.&.&.\end{bmatrix},\begin{bmatrix}1&1&1&.&.\\.&1&1&.&.\\.&.&1&5&1\\.&.&.&5&-1\\.&.&.&.&5\end{bmatrix}\right)\cdot\to(\mathrm{k0,m0})';$$

*) $(k_V[i,j], m_V[i,j]) = (k_0[I(i),J(j)], m_0[I(i), J(j)])$ –>

$$.\to\cdot\left(\begin{bmatrix}1&-1&.&.&.\\2&1&-1&.&1\\3&2&1&5&.\\5&5&.&.&.\\\bullet&-1&2&.&.\end{bmatrix},\begin{bmatrix}1&1&1&.&.\\.&1&1&.&.\\.&.&1&5&1\\.&.&.&.&5\\.&.&.&5&-1\end{bmatrix}\right).$$

Потрібно знову нагадати, що саме у Virtual-формі ($k_V[i,j]$, $m_V[i,j]$) імітується процес приведення до каноніки. Видно, що у віртуальній формі в позиції $k_V[5,1]$ з'явився нуль. Ну а фізично в пам'яті PC зберігається тільки ($k_0[i,j]$, $m_0[i,j]$) – (Original-форма), яка не є канонічною.

В матриці v^T формується Transformer лівого перетворення:

*) $S_{I(5),J(4)}\cdot v^T[i, j]$·–> $v^T[i,j]'$.

Тепер бачимо, що після лівого перетворення пучка, в трикутній формі матриці m_V з'явилось пошкодження в елементі [5, 4]. Для відновлення трикутної форми m_V , використовуються праві Transformer.

Вилучається позиція [5, 4] з допомогою правого перетворення S_{54}. Тепер працює стратегія вибору ведучого елемента з двох сусідніх позицій 5-го рядка m_V: [5, 4] і [5, 5]. З метою врегулювання величини s_{54} змінюємо конфігурацію вектора Pointer J, що відповідає перестановці 4-го та 5-го стовбців:

$J = \{1, 2, 3, 5, 4\}$; s_{54} = 1/5.

Original та Virtual форми змінюються за формулами правого перетворення:

*) $(k_0, m_0)\ S_{J(5),J(4)}$ –>

$$\to\left(\begin{bmatrix}1&-1&.&.&.\\2&1&-1&.&1\\3&2&1&5&.\\\bullet&-1&2&.&.\\5&5&.&.&.\end{bmatrix},\begin{bmatrix}1&1&1&.&.\\.&1&1&.&.\\.&.&1&5&1\\.&.&.&5&-1\\.&.&.&.&5\end{bmatrix}\right)\cdot\begin{bmatrix}1&.&.&.&.\\.&1&.&.&.\\.&.&1&.&.\\.&.&.&1&1/5\\.&.&.&.&1\end{bmatrix}\cdot\to$$

$$\to\cdot\left(\begin{bmatrix}1&-1&.&.&.\\2&1&-1&.&1\\3&2&1&5&1\\\bullet&-1&2&.&.\\5&5&.&.&.\end{bmatrix},\begin{bmatrix}1&1&1&.&.\\.&1&1&.&.\\.&.&1&5&2\\.&.&.&5&\bullet\\.&.&.&.&5\end{bmatrix}\right)\cdot\to(\mathrm{k0},\mathrm{m0})';$$

*) $(k_V[i,j], m_V[i,j]) = (k_0[I(i),J(j)], m_0[I(i), J(j)])$ –>

$$\cdot\to\cdot\left(\begin{bmatrix}1&-1&.&.&.\\2&1&-1&1&.\\3&2&1&1&5\\5&5&.&.&.\\\bullet&-1&2&.&.\end{bmatrix},\begin{bmatrix}1&1&1&.&.\\.&1&1&.&.\\.&.&1&2&5\\.&.&.&5&.\\.&.&.&\bullet&5\end{bmatrix}\right)\cdot.$$

В матриці u формується Transformer правого перетворення-:

*) $u[i,j]\cdot S_{J(5),J(4)}\cdot$–> $u[i,j]'$.

На цьому закінчується перший крок. Результат – видалена позиція $k_V[5,1]$ при збереженні трикутної форми m_V.

в) На другому і наступних кроках дотримуємось умов циклічності. Вилучається позиція $k_V[4,1]$ з допомогою лівого перетворення S_{43}. Використовується стратегія вибору ведучого елемента з двох сусідніх позицій 1-го стовбця: [4, 1] і [3, 1]. З метою врегулювання величини s_{43} змінюється конфігурація вектора Pointer I, що відповідає перестановці 3-го та 4-го рядків матриць:

$$I = \{1, 2, 5, 3, 4\};\quad s_{43} = -3/5.$$

Original та Virtual форми пучка змінюються за формулами лівого перетворення:

*) $S_{I(4),I(3)}\cdot(k_0, m_0)$–> $(k_0, m_0)'$;

*) $(k_V[i,j], m_V[i,j]) = (k_0[I(i),J(j)], m_0[I(i), J(j)])$.

В матриці v^T формується Transformer:

*) $S_{I(4),I(3)}\cdot v^T[i,j]\cdot$–> $v^T[i,j]'$.

Для відновлення трикутної форми m_V використовується правий Transformer S_{43}. Працює стратегія вибору ведучого елемента з двох сусідніх позицій 4-го рядка m_V: [4, 3] і [4, 4]. Конфігурацію вектора Pointer J не змінюємо:

$$J = \{1, 2, 3, 5, 4\};\quad s_{43} = 1.$$

Original та Virtual форми змінюються за формулами правого перетворення:

*) $(k_0, m_0)\, S_{J(4),J(3)}$ –> $(k_0, m_0)''$;

*) $(k_V[i,j], m_V[i,j]) = (k_0[I(i),J(j)], m_0[I(i), J(j)])$

В матриці u формується правий Transformer:

*) $u[i,j]\cdot S_{J(4),J(3)}\cdot$–> $u[i,j]'$.

Порядок операцій такий, що на кожному кроці зберігаються всі нульові елементи, що одержані на попередніх.

Приведемо проміжні результати перетворення пучка після вилучення ненульових позицій 1-го стовбця k_V .

-) $I(i)=\{1, 5, 2, 3, 4\}$; $J(j)=\{1, 2, 3, 5, 4\}$;

$$\cdot(\mathrm{k0},\mathrm{m0})=\left(\begin{bmatrix}1&-1&.&.&.\\\bullet&-1&.&.&1\\\bullet&1&2&5&1\\\bullet&1&2&.&.\\5&5&.&.&.\end{bmatrix},\begin{bmatrix}1&2&1&.&.\\.&.&-1&.&-2\\.&.&.&5&-1\\.&.&.&5&.\\.&5&5&.&5\end{bmatrix}\right)\cdot;$$

$$(k_V,m_V)=\left(\begin{bmatrix}1&-1&.&.&.\\5&5&.&.&.\\\bullet&-1&.&1&.\\\bullet&1&2&1&5\\\bullet&1&2&.&.\end{bmatrix},\begin{bmatrix}1&2&1&.&.\\.&5&5&5&.\\.&.&-1&-2&.\\.&.&.&-1&5\\.&.&.&.&5\end{bmatrix}\right);$$

г) Після останнього кроку циклічного процесу буде одержана вся необхідна інформація приведення пучка до узагальненої форми Хесенберга. Проведемо аналіз одержаних результатів.

-) $I(i) = \{1, 5, 2, 3, 4\}$; $J(j) = \{1, 2, 3, 5, 4\}$;

$$\cdot(\mathrm{k0},\mathrm{m0})=\left(\begin{bmatrix}1&-1&.&.&.\\\bullet&-1&.&.&1\\\bullet&\bullet&3&5&2\\\bullet&\bullet&\bullet&-10/3&-1/3\\5&5&.&.&.\end{bmatrix},\begin{bmatrix}1&2&1&.&.\\.&.&-1&.&-2\\.&.&.&5&-3\\.&.&.&5/3&.\\.&5&5&.&5\end{bmatrix}\right)\cdot;$$

Для перевірки проведених обчислень знайдемо приведення стартового пучка з допомогою лівого та правого фінальних Transformer:

*) $v^T\cdot(k, m)\cdot u\to$

$$\to\begin{bmatrix}1&.&.&.&.\\.&1&.&.&-2/5\\.&1&1&.&-1\\.&1/3&-2/3&1&-8/15\\.&.&.&.&1\end{bmatrix}*\left(\begin{bmatrix}1&-1&.&.&.\\2&1&-1&.&1\\3&2&1&5&.\\4&3&2&.&.\\5&5&.&.&.\end{bmatrix},\begin{bmatrix}1&1&1&.&.\\.&1&1&.&.\\.&.&1&5&1\\.&.&.&5&3\\.&.&.&.&5\end{bmatrix}\right)*$$

$$\begin{bmatrix}1&.&.&.&.\\.&1&.&.&.\\.&1&1&.&.\\.&1/5&2/5&1&1/5\\.&1&1&.&1\end{bmatrix}\to(\mathrm{k0},\mathrm{m0})$$

Результати співпадають. Це підтвердження правильності попередніх обчислень.

*) Наостанок, канонічну форму (узагальнену Хесенберга) можна відтворити згідно (3). Потрібно переставити рядки/стовбці (k_0, m_0) як вказано в векторах-Pointer (I, J):

$$(k_V,m_V)=\left(\begin{bmatrix}1&-1&.&.&.\\5&5&.&.&.\\\bullet&-1&.&1&.\\\bullet&\bullet&3&2&5\\\bullet&\bullet&\bullet&-1/3&-10/3\end{bmatrix},\begin{bmatrix}1&2&1&.&.\\.&5&5&5&.\\.&.&-1&-2&.\\.&.&.&-3&5\\.&.&.&.&5/3\end{bmatrix}\right);$$

Знову звернемо увагу на матриці-Transformer v^T, u – не випадково, що всі їх елементи обмежені величиною 1.

Таким чином, і в цьому прикладі не применшуючи загальності виконане потрібне перетворення заданого пучка матриць, а також доказ цієї теореми.

6. Висновки. В роботі запропоновано схему організації обчислень задач матричної алгебри, при якій узгоджуються компромісні для обчислень питання точності та стійкості алгоритму, економії оперативної пам'яті, раціональної кількості операцій та інші. В основі алгоритму лежить ряд взаємопов'язаних між собою положень:

- Базовою в цих перетвореннях прийнята елементарна неунітарна матриця типу S, яка виконує лише пряму функцію − вилучення ненульових позицій.
- Для перетворень матричних рівнянь використовуються лише матриці типу Transformer − ланцюги, що складаються з матриць типу S, які добре обумовлені.
- Величини всіх елементів матриць типу S в процесі обчислень обмежені величиною 1, чим стримується необмежений ріст величин проміжних обчислень.
- Такі обмеження на проміжні обчислення забезпечуються в схемі відповідними перестановками елементів.
- Для забезпечення раціональної кількості операцій виконуються перестановки не самих елементів матриць, а номерів їх рядків/стовбців. З цією метою введено апарат вказівників-Pointer.
- Для раціонального використання оперативної пам'яті PC та управління роботою введено механізм Virtual та Original матриць, коли в PC зберігається лише один масив.
- Для взаємозв'язку між введеними елементами схеми приведені аналітичні залежності.
- Не видно явних обмежень при практичних обчисленнях, які б застерігали від роботи з широким діапазоном чисельних значень коефіцієнтів.
- Перетворення в конкретних задачах мають форму ланцюгів з елементарних матриць, формують циклічні процеси та мають характер універсальних.
- В роботі наведені приклади важливих для матричної алгебри задач.

Список літератури

1. *Икрамов Х.Д.* Численное решение матричных уравнений. Ортогональные методы. Москва: Наука. 1984. 192 с.

2. *Мальцев А.И.* Основы линейной алгебры. Москва: Гостехиздат, 1956.

3. *Гантмахер Ф.Р.*Теория матриц. Москва: Гостехиздат, 1967. 575 с.

4. *Уилкинсон, Райнш* Справочник алгоритмов на языке Алгол. Линейная алгебра. Москва: Машиностроение. 1976. 389 с.

5. *Бате К., Вилсон Е.* Численные методы анализа и метод конечных элементов. Москва: Стройиздат, 1982. 448 с.

6. *Воеводин В.В.* Вычислительные основы линейной алгебры. Москва: Наука, 1977. 304 с.

7. *Воеводин В.В., Кузнецов Ю.А.* Матрицы и вычисления. Москва: Наука, 1984. 320 с.

8.. *Ланкастер П.* Теория матриц. Москва: Наука, 1978. 280 с.

9. *Постнов В.А, Херхурим И.Я.* Метод конечных элементов в расчетах судовых конструкций. Ленинград: Судостроение, 1974.

10. *Якубович В.А., Старжинский В.М.* Линейные дифференциальные уравнения с периодическими коэффициентами и их приложения. Москва: Наука, 1972, 720 с.

11. *Форсайт Дж, Малькольм М., Моулер К.* Машинные методы математических вычислений. Москва: Мир, 1980. 280 с.

12. *Беклемишев Д.В.* Дополнительные главы линейной алгебры. Москва: Наука, 1983. 336 с.

13. *Демидович Б.П., Марон И.А.* Основы вычислительной математики. Москва: Наука. 1970. 664 с.

Bibliography (transliterated)

1. *Ikramov H.D.* Chislennoe reshenie matrichnyh uravnenij. Ortogonal'nye metody. Moscow: Nauka. 1984. 192 p.

2. *Mal'cev A.I.* Osnovy linejnoj algebry. Moscow: Gostehizdat, 1956.

3. *Gantmaher F.R.*Teoriya matric. Moscow: Gostehizdat, 1967. 575 p.

4. *Uilkinson, Rajnsh* Spravochnik algoritmov na yazyke Algol. Linejnaya algebra. Moscow: Mashinostroenie. 1976. 389 p.

5. *Bate K., Vilson E.* Chislennye metody analiza i metod konechnyh elementov. Moscow: Strojizdat, 1982. 448 p.

6. *Voevodin V.V.* Vychislitel'nye osnovy linejnoj algebry. Moscow: Nauka, 1977. 304 p.

7. *Voevodin V.V., Kuznecov Yu.A.* Matricy i vychisleniya. Moskva: Nauka, 1984. 320 p.

8.. *Lankaster P.* Teoriya matric. Moscow: Nauka, 1978. 280 p.

9. *Postnov V.A, Harhurim I.Ya.* Metod konechnyh elementov v raschetah sudovyh konstrukcij. Leningrad: Sudostroenie, 1974.

10. *Yakubovich V.A., Starzhinskij V.M.* Linejnye differencial'nye uravneniya s periodicheskimi koefficientami i ih prilozheniya. Moscow: Nauka, 1972, 720 p.

11. *Forsajt Dzh, Mal'kol'm M., Mouler K.* Mashinnye metody matematicheskih vychislenij. Moscow: Mir, 1980. 280 p.

12. *Beklemishev D.V.* Dopolnitel'nye glavy linejnoj algebry. Moscow: Nauka, 1983. 336 p.

13. *Demidovich B.P., Maron I.A.* Osnovy vychislitel'noj matematiki. Moscow: Nauka. 1970. 664 p.

Надійшла (received) 24.09.2020

Відомості про авторів / Сведения об авторах / About the Authors

Грищенко Володимир Миколайович (***Грищенко Владимир Николаевич, Grischenko Volodymir Mykolayovich***) – кандидат технічних наук, доцент кафедри динаміки та міцності машин, НТУ «ХПІ». Тел.: 707 68 79. E-mail: grivn_dmm@ukr.net