УДК 378.016+ 512.643

Методические аспекты автоматической генерации задач по линейной алгебре

© Я.Ю. Коновалов, С.К. Соболев, М.А. Ермолаева

МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

В статье предлагаются алгоритмы генерирования широкого спектра задач по линейной алгебре: задание матрицы любого размера с заданным определителем, с заданным рангом; матричных уравнений, систем линейных уравнений, в том числе с параметром, задач на линейные операторы и квадратичные формы. Алгоритмы успешно опробованы на практике, с помощью них может быть составлено неограниченное число вариантов контрольных работ, проводимых кафедрой «Высшая математика» в первом семестре в модуле «Матрицы и СЛАУ», а также вариантов индивидуальных заданий и рубежного контроля по модулю «Линейная алгебра». Отдельным файлом одновременно генерируются все ответы, в некоторых случаях и промежуточные.

Ключевые слова: *матрица, автоматическая генерация, определитель, ранг, случайные числа, обратная матрица, система линейных уравнений, параметр, векторы, базис, решение.*

Введение. Проблема автоматической генерации задач не нова и возникла намного раньше массового появления компьютеров. Известно, что современные студенты очень быстро обзаводятся ответами и даже полными решениями (иногда с ошибками) имеющихся комплектов контрольных работ (КР) уже после первого их использования преподавателем, а уж про индивидуальные домашние задания (ДЗ), висящие на стендах по нескольку лет (а иногда и десятилетий) и говорить нечего. Кроме того, на кафедрах регулярно возникает потребность обновления условий КР или ДЗ по чисто методическим причинам: надо упростить (или усложнить) условия задач, добавить задачи нового типа, да и просто в связи с появлением новых дисциплин или новых разделов старых.

Итак, возникает необходимость придумать по каждой теме не одну, а 10–15 вариантов задач для КР и даже 30 вариантов для ДЗ, различных, но относительно однотипных и сравнимых по сложности, обязательно с ответами и желательно хотя бы с краткими решениями. В идеале комплекты по 10–30 задач хорошо бы в разных группах сделать разными. В каждом варианте КР может быть от трех до пяти различных задач, а в ДЗ их число может доходить и до 10–15. Естественно, при их создании должна быть минимизирована вычислительная работа преподавателя и максимально задействованы современные ресурсы персональных компьютеров.

Составлять задачи надо так, чтобы ответы к ним были **по возможности** «хорошими», например, корни характеристического урав-

нения были целыми, а не, скажем $\frac{-13 \pm \sqrt{137}}{19}$. Кроме того, возникает много других «методических» ограничений, на которых мы подробно остановимся ниже.

В литературе проблема генерирования задач по математике обсуждается давно [1 — 4]. Отметим, что многие реально работающие генераторы устроены так: имеется большой банк уже готовых задач на разные темы, и при генерировании варианта, состоящего из набора задач на заданные темы, задачи из банка выбираются случайно.

В настоящей работе описаны результаты создания генератора именно самих задач с ответами (окончательными и промежуточными). Мы ограничились задачами, так или иначе связанными с матрицами: ранг матрицы и системы векторов, обратная матрица, матричные уравнения, системы линейных алгебраических уравнений (СЛАУ), в том числе и с параметром, приведение матрицы линейного оператора к диагональному виду, приведение квадратичной формы к каноническому виду с помощью ортогонального преобразования.

Широко известны способы составления задач «от обратного», когда условие задачи строится по заданному ответу. Например, квадратное уравнение с корнями x_1 и x_2 можно составить в виде $a(x-x_1)(x-x_2)$.

Для решения задач, связанных с матрицами и системами линейных уравнений (СЛАУ), широко применяется метод элементарных преобразований. Используя его в способе «от обратного» можно генерировать условия задач при помощи относительно простых алгоритмов.

Напомним, что элементарными преобразованиями матрицы называются прибавление к одной строке (столбцу), матрицы другой строки (столбца), умноженной на некоторое число, умножение строки (столбца) матрицы на любое число, не равное нулю, перестановка строк (столбцов). Известно [5–8], что при элементарных преобразованиях не меняется ранг матрицы, а при первом из преобразований у квадратных матриц сохраняется определитель. Кроме того, при элементарных преобразованиях строк расширенной матрицы СЛАУ не меняется множество ее решений.

Ниже описаны алгоритмы, которые успешно применены на практике для автоматической генерации задач и формирования из них вариантов КР и ДЗ по аналитической геометрии и линейной алгебре.

1. Алгоритм генерации матриц с заданным определителем. Пусть требуется построить матрицу размера $n \times n$ с определителем d. Возьмем верхнетреугольную матрицу A, у которой один из диагональных элементов равен d, а остальные 1 (звездочкой обозначены произвольные случайные числа).

$$A = \begin{pmatrix} d & & & * \\ 0 & 1 & & \\ & & \ddots & \\ & 0 & & 1 \end{pmatrix}. \qquad (1)$$

Ее определитель равен d. Подвергнем матрицу (1) серии случайных преобразований, прибавляя к случайно выбранной строке (столбцу) другую случайно выбранную строку (столбец), умноженную на случайное целое ненулевое число. В дальнейшем будем называть такую процедуру ***перемешиванием строк (столбцов)***. Полученная в результате матрица также будет иметь определитель d. При этом требуется, чтобы, с одной стороны, элементы матрицы были не очень большими, с другой стороны, матрица не должна содержать избыточного количества нулей. Наиболее простым способом обеспечить выполнение этих требований является проверка на выходе и генерация новой матрицы в случае, если текущая им не удовлетворяет. Поскольку как сама генерация, так и проверка выполняются очень быстро, алгоритм в целом получается эффективным.

Отметим, что такая структура алгоритма, видимо, является общей для всех генераторов задач, так как учесть все возможности в процессе составления условий невозможно. Однако, несмотря на кажущуюся неэффективность, она имеет свои преимущества. Ведь при таком подходе нет необходимости придумывать способ генерации, гарантирующий качество полученного результата. Достаточно лишь предложить алгоритм, выдающий хорошие условия, относительно часто и грамотно организовать проверку, чтобы отсечь неудачные.

2. Алгоритм генерации матричных уравнений. Рассмотрим алгоритм генерации матричных уравнений вида $AX = B$, где A и B — известные квадратные матрицы, а X — искомая. При решении студент должен обратить матрицу A по формуле $A^{-1} = \frac{1}{\det A}\tilde{A}$, где $\tilde{A} = (A_{ij})^T$ — присоединенная матрица (т. е. составленная из алгебраических дополнений матрицы A и затем транспонированная), и найти $X = A^{-1}B$. При этом матрица A должна быть невырожденной и иметь небольшой определитель, но не равный единице, чтобы студент, забывший на него поделить, не мог случайно получить правильный ответ. Также матрица A не должна быть симметричной, чтобы студент не мог получить правильный ответ в случае, если он забыл транспонировать матрицу алгебраических дополнений, что является довольно распространенной ошибкой.

Для генерации квадратной матрицы A (порядка n) зададим ее определитель небольшим случайным числом, отличным от нуля и

единицы, и применим алгоритм построения матрицы с заданным определителем, описанный выше. Выполнение требования несимметричности матрицы A обеспечивается дополнительной проверкой. Матрицу X (размером $n \times m$, не обязательно квадратную) составляем из случайных чисел, находим B как $B = AX$ и проверяем ее на наличие избыточно больших элементов. Аналогичным образом составляем матричные уравнения вида $XA = B$ и $AXB = C$. В последнем случае матрицы X и C имеют одинаковые размеры.

3. Задачи на ранг и линейную зависимость. Алгоритм генерации матричных уравнений можно применить для генерации матриц с заданным рангом r. Для этого составим матрицу A, состоящую из единичной матрицы E порядка r, прямоугольной матрицы D случайных чисел $k_{i,j}$ размера $r \times (m-r)$ и $(n-r)$ нулевых строк:

$$A = \begin{pmatrix} E & D \\ O & O \end{pmatrix} = \left(\begin{array}{ccc|c} 1 & & 0 & \\ & \ddots & & k_{i,j} \\ 0 & & 1 & \\ \hline & 0 & & 0 \end{array}\right). \qquad (2)$$

Ранг этой матрицы равен r. Подвергнем строки матрицы перемешиванию. Полученная матрица также имеет ранг r. После этого имеет смысл поделить каждую строку на наибольший общий делитель ее элементов. Это увеличит вероятность того, что матрица пройдет проверку на максимальное значение элементов. Кроме того, необходимо проверить количество нулей и убедиться в отсутствии пропорциональных строк или столбцов, так как их наличие существенно упрощает задачу нахождения ранга. Если задачу планируется решать методом окаймляющих миноров, имеет смысл произвести перестановку столбцов, распределив первые r столбцов по матрице, чтобы усложнить поиск базисного минора.

В контрольных работах по теме «Матрицы и СЛАУ» в МГТУ им. Н.Э. Баумана, а также в [9] присутствует задача о нахождении ранга матрицы в зависимости от параметра. Рассмотрим алгоритм ее генерации. Модифицируем матрицу (2), добавив в $(r+1)$-ю строчку на место с номером $q > r$ элемент $\lambda - \lambda_0$. Получим

$$\left(\begin{array}{ccc|ccccc} 1 & & 0 & & & & & \\ & \ddots & & & & k_{i,j} & & \\ 0 & & 1 & & & & & \\ \hline & 0 & & 0 & \cdots & \lambda-\lambda_0 & \cdots & 0 \\ \hline & 0 & & & & 0 & & \end{array}\right). \qquad (3)$$

Ранг матрицы (3) равен r при $\lambda = \lambda_0$ и $r+1$ при $\lambda \neq \lambda_0$. Прибавим к $(r+1)$-й строке случайную линейную комбинацию первых r (базисных) строк. Подвергнем все строки кроме $(r+1)$-й перемешиванию. Помимо перечисленных выше проверок строка и столбец полученной матрицы, содержащие параметр λ, не должны становиться пропорциональными другим строкам и столбцам после выбрасывания элементов, стоящих на месте q (для строк) и $r+1$ (для столбцов) соответственно. В противном случае найти λ_0 окажется очень легко.

4. Алгоритм генерации СЛАУ. Способ генерации матриц с заданным рангом *r* лежит в основе алгоритма генерации задач на решение СЛАУ.

Пусть требуется составить СЛАУ из n строк и m столбцов ранга r $(r \le m,\ r \le n)$. Построим матрицу системы вида (2). Добавим случайный столбец *B* в правой части:

$$\left(\begin{array}{ccc|c} 1 & & 0 & b_1 \\ & \ddots & & \vdots \\ & & & \\ 0 & & 1 & b_r \\ \hline & 0 & & 0 \end{array}\right) = \left(\begin{array}{c|c|c} E & D & B \\ \hline O & O & O \end{array}\right). \qquad (4)$$

Легко показать, что решение этой системы уравнений имеет вид

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} = \sum_{j=1}^{m-r} c_j \begin{pmatrix} -\mathbf{k}_j \\ 0 \\ \vdots \\ 1_{r+j} \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_r \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \left(\begin{array}{c} -D \\ \hline E \end{array}\right) C + \left(\begin{array}{c} B \\ \hline O \end{array}\right), \qquad (5)$$

где матрица *D* состоит из столбцов $\mathbf{k}_j = \begin{pmatrix} k_{1j} \\ \vdots \\ k_{rj} \end{pmatrix}$, $C = \begin{pmatrix} c_1 \\ \dots \\ c_{m-r} \end{pmatrix}$ — столбец произвольных констант, а 1_{r+j} означает, что элемент 1 стоит на $(r + j)$-м месте. Затем, аналогично предыдущему алгоритму подвергнем матрицу перемешиванию строк. Поскольку при элементарных преобразованиях решение СЛАУ не меняется, ответ по-прежнему задается формулой (5). Для дополнительного усложнения задачи можно переставить столбцы в левой части СЛАУ (и соответственно стро-

ки в ответе) так, чтобы базисные столбцы шли не подряд. Для генерации **однородной** СЛАУ задаем $b_i = 0$. Как и в задачах на ранг матрицы, необходимо проверить отсутствие пропорциональных строк и столбцов, а также избытка нулей.

Отметим, что при генерации по этому алгоритму полной квадратной СЛАУ определитель ее матрицы окажется равным 1. Поэтому построение СЛАУ, которые предполагается решать методом Крамера, рекомендуется осуществлять аналогично описанной выше процедуре генерации матричных уравнений $AX = B$, заменив искомую матрицу X и матрицу правой части B на столбец неизвестных X и столбец свободных членов B соответственно.

Обобщим алгоритм для генерации СЛАУ, зависящих от параметра. В зависимости от положения параметра и его влияния на решение рассмотрим четыре случая.

Первый случай: параметр **в правой части влияет** на существование решения.

Аналогично предыдущему составим СЛАУ вида (4). Зададим последнюю строку как случайную линейную комбинацию первых r (базисных) строк. Добавим к свободному члену последней строки $\lambda - \lambda_0$, где λ_0 – случайное число. Оставшиеся строки аналогично подвергнем перемешиванию, не затрагивая последнюю строку. Полученная СЛАУ будет иметь решение (5) при $\lambda = \lambda_0$ и будет несовместна при $\lambda \neq \lambda_0$.

Второй случай: параметр **в правой части не влияет** на существование решения. Составим СЛАУ вида (4), умножим некоторую базисную строку (например, *r*-ю) на случайное число $q \neq 0$ и прибавим к ее свободному члену параметр λ. Получим СЛАУ следующего вида:

$$\left(\begin{array}{ccc|c} 1 & & & b_1 \\ & \ddots & k_{ij} & \vdots \\ & 1 & & b_{r-1} \\ \hline & q & qk_{rj} & qb_r + \lambda \\ \hline & 0 & 0 & 0 \end{array}\right). \qquad (6)$$

Прибавим к r-й строке случайную линейную комбинацию предыдущих базисных строк. Затем подвергнем перемешиванию все строки кроме r-й (для этого удобно поставить ее на последнее место). СЛАУ (6) и соответственно полученная СЛАУ при любом значении λ будут иметь следующее решение:

$$X = \sum_{j=1}^{m-r} c_j \begin{pmatrix} -\mathbf{k}_j \\ 0 \\ \vdots \\ 1_{r+j} \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_{r-1} \\ b_r + \dfrac{\lambda}{q} \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \qquad (7)$$

Третий случай: параметр **в левой части уравнения влияет** на существование решения. В матрице вида (4) поставим элемент $\lambda - \lambda_0$ в $(r+k)$-й столбец $(r+1)$-й строчки и добавим в правую часть элемент $b_{r+1} \neq 0$, получим

$$\left(\begin{array}{ccc|c|c} 1 & & 0 & & b_1 \\ & \ddots & & k_{ij} & \vdots \\ 0 & & 1 & & b_r \\ \hline 0 & & 0 \;\cdots\; \lambda-\lambda_0 \;\cdots\; 0 & & b_{r+1} \\ \hline 0 & & 0 & & 0 \end{array}\right). \qquad (8)$$

Аналогично предыдущему добавим к этой строчке случайную линейную комбинацию базисных строк, а остальные строки подвергнем перемешиванию. Полученная СЛАУ будет несовместна при $\lambda = \lambda_0$. При $\lambda \neq \lambda_0$ $x_{r+k} = -\dfrac{b_{r+1}}{\lambda - \lambda_0}$, k-й столбец (5) переходит в свободный член и решение (8) представляется в виде (9).

$$X = \sum_{\substack{j=1 \\ j\neq k}}^{m-r} c_j \begin{pmatrix} -\mathbf{k}_j \\ 0 \\ \vdots \\ 1_{r+j} \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_r \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \frac{b_{r+1}}{\lambda - \lambda_0} \begin{pmatrix} \mathbf{k}_k \\ 0 \\ \vdots \\ -1_{r+k} \\ \vdots \\ 0 \end{pmatrix}. \qquad (9)$$

Четвертый случай: параметр в **левой части уравнения влияет** на размерность системы решений. В матрице вида (4) поставим элемент $\lambda - \lambda_0$ в $r+k$-й столбец $r+1$-й строчки. Получим (8) с $b_{r+1} = 0$. Добавим к $r+1$-й строчке случайную линейную комбинацию базисных строк, а остальные строки подвергнем перемешиванию. При

$\lambda = \lambda_0$ СЛАУ равносильна (4) и ее решение есть (5). При $\lambda \neq \lambda_0$ добавляется условие $x_{r+k} = 0$, что приводит к выпадению из (5) k-го столбца.

Отметим, что описанный выше алгоритм генерации условия задачи на нахождение ранга матрицы в зависимости от параметра аналогичен четвертому случаю алгоритма генерации СЛАУ с параметром.

Помимо собственно генерации СЛАУ приведенные алгоритмы генерируют системы векторов и матрицы заданного ранга, что позволяет использовать их при составлении различных задач на линейную зависимость, и базисы. Кроме того, определители и СЛАУ широко используются при решении многих задач аналитической геометрии, следовательно, описанные алгоритмы применимы и для них.

5. Задачи на собственные значения. В курсе линейной алгебры ключевое положение занимают задачи нахождения собственных векторов и собственных значений линейных операторов и квадратичных форм. Опишем алгоритмы их генерации.

Известно [5–9], что для любого линейного оператора в пространстве $\mathbb{R}^n$ найдется базис, в котором матрица линейного оператора имеет жорданов вид, а матрица оператора в исходном базисе равна $A = PJP^{-1}$, где J – матрица оператора в жордановой форме, а P – матрица перехода.

Для генерации задачи о нахождении собственных значений оператора в трехмерном вещественном пространстве построим матрицу оператора как $A = PJP^{-1}$. Жорданова форма такого оператора может быть одного из следующих четырех видов:

$$\begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}, \quad \begin{pmatrix} \lambda_1 & 1 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_2 \end{pmatrix}, \quad \begin{pmatrix} \lambda_1 & 1 & 0 \\ 0 & \lambda_1 & 1 \\ 0 & 0 & \lambda_1 \end{pmatrix}, \quad \begin{pmatrix} \alpha & \beta & 0 \\ -\beta & \alpha & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}.$$

В первом случае трем собственным значениям соответствуют три собственных вектора — столбцы $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ матрицы перехода P; во втором случае двум собственным значениям соответствуют собственные векторы $\mathbf{p}_1$ и $\mathbf{p}_3$; в третьем — единственному собственному значению λ_1 соответствует вектор $\mathbf{p}_1$; четвертый случай соответствует наличию одного вещественного собственного значения λ_3 с собственным вектором $\mathbf{p}_3$ и пары комплексных $\lambda_{1,2} = \alpha \pm i\beta$. В зависимости от требований конкретной задачи выбор случаев можно ограничить. Например, для задачи о приведении оператора к диагональному виду нужно рассматривать только первый случай. Выберем один из случаев и зададим целые λ_i случайным образом. Поскольку

обычно первый корень кубического многочлена ищут подбором, по крайней мере, одно из собственных значений должно быть целым числом в пределах от -3 до 3. В качестве матрицы P подойдет любая невырожденная матрица. При вычислении P^{-1} в знаменателе оказывается $\det P$, и элементы матрицы A могут оказаться не целыми, что значительно усложнит задачу. Бороться с этим явлением можно двумя способами, либо взять матрицу P с определителем, равным единице (алгоритм составления таких матриц приведен выше), либо выбирать все или некоторые λ_i, кратные $\det P$. Полученную матрицу проверяем на наличие избыточно больших элементов и нулей. Кроме того, полезно проверить характеристический многочлен на наличие больших коэффициентов.

Описанный способ без существенных изменений может быть обобщен для построения матриц операторов большей размерности. Наибольшую трудность здесь представляет составление таких задач, чтобы найти корни характеристического многочлена было не очень трудно.

Важным преимуществом автоматической генерации также является возможность составления подробных ответов, что обычно является весьма трудоемкой задачей при составлении задач вручную. В частности, в описанной задаче можно не только указать собственные векторы и собственные значения, но и явно выписать характеристический многочлен, чтобы преподаватель легко мог обнаружить ошибку при его составлении.

6. Квадратичные формы. Матрицы квадратичных форм преобразуются по формуле $A' = P^T AP$, где A и A' — матрицы квадратичной формы в старом и новом базисе соответственно, но в отличие от операторов часто требуется, чтобы матрица P была ортогональной: $P^T = P^{-1} \Leftrightarrow P^T P = E$. В этом случае $A = PA'P^T$. Некоторые методы генерирования ортогональных матриц любого размера описаны в конце статьи и в [10].

Генерация задачи о построении **кривой второго порядка**, заданной в виде квадратичной формы, не представляет трудности. Составим матрицу квадратичной формы в виде PDP^T, где D – диагональная матрица, а P – ортогональная матрица. Для двумерного случая матрица

$$P = \frac{1}{a^2 + b^2}\begin{pmatrix} a & b \\ -b & a \end{pmatrix},$$

все вычисления можно провести в целых числах. Компенсировать деление на определитель можно умножением на него собственных значений.

Аналогично организуем генерацию условий задачи об исследовании знакоопределенности квадратичной формы от двух переменных в зависимости от значения параметра. Составим ее матрицу в виде PDP^T, где D — диагональная матрица, на диагонали у которой стоят многочлены первой степени, а P — обратная матрица перехода. Практика показала, что относительно небольшие коэффициенты чаще всего получаются в случае, когда столбцы матрицы P ортогональны, хотя в качестве P подойдет любая невырожденная матрица. Формально перемножая матрицы, получим требуемую квадратичную форму. Аналогично можно получить квадратичные формы более высокой размерности для случаев, в которых не требуется ортогональность преобразования P, например для исследования знакоопределенности квадратичной формы.

К сожалению, применение приведенного выше алгоритма для генерации матриц квадратичных форм в случае, когда матрица P должна быть ортогональной, затруднительно. Например, в задаче о приведении квадратичной формы к диагональному виду ортогональным преобразованием оказывается проще рассмотреть частный случай, дающий тем не менее достаточно обширный набор задач.

Для генерации условий задачи о диагонализации квадратичной формы ортогональным преобразованием рассмотрим три квадратичные формы:

$$A=\begin{pmatrix} a^2 & ab & ac \\ ab & b^2 & bc \\ ac & bc & c^2 \end{pmatrix}, \qquad B=\begin{pmatrix} b^2 & -ab & 0 \\ -ab & a^2 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad C=\begin{pmatrix} d & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & d \end{pmatrix}.$$

Все они имеют собственные векторы

$$\mathbf{e}_1=\begin{pmatrix} a \\ b \\ c \end{pmatrix}, \qquad \mathbf{e}_2=\begin{pmatrix} b \\ -a \\ 0 \end{pmatrix}, \qquad \mathbf{e}_3=\begin{pmatrix} ac \\ bc \\ -a^2-b^2 \end{pmatrix}$$

со следующими наборами собственных значений:
для A: $\left(a^2+b^2+c^2,\ 0,\ 0\right)$, для B: $\left(0,\ a^2+b^2,\ 0\right)$, для C: $\left(d, d, d\right)$. Тогда их линейная комбинация $\alpha A+\beta B+C$ с теми же собственными векторами будет иметь собственные значения $\alpha\left(a^2+b^2+c^2\right)+d$ для $\mathbf{e}_1$, $\beta\left(a^2+b^2\right)+d$ для $\mathbf{e}_2$ и d для $\mathbf{e}_3$ соответственно. К полученной квадратичной форме предъявляются те же требования, что и к оператору: не очень большие коэффициенты матрицы и характеристического многочлена, одно из собственных значений по модулю меньше либо

равно трем. Последнее требование легко обеспечить уже на этапе генерации, выбирая подходящие α, β и d.

7. Генерация ортогональных матриц произвольного порядка.

1-й метод. Каждую ортогональную матрицу Q порядка n можно представить в виде произведений простейших матриц поворота $R_{ij}(\varphi)$ в плоскости $x_i x_j$, где $1 \le i < j \le n$, $-\pi \le \varphi \le \pi$:

$$R_{ij}(\varphi) = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & \cos\varphi & \cdots & 0 & \cdots & -\sin\varphi & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & \sin\varphi & \cdots & 0 & \cdots & \cos\varphi & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \begin{matrix} \\ \\ i \\ \\ \\ \\ j \\ \\ \\ \end{matrix}.$$

$$\begin{matrix} & & i & & & & j & & \end{matrix}$$

Таким образом, всякая ортогональная матрица Q есть произведение $\frac{1}{2}n(n-1)$ матриц такого вида, где все углы φ_{ij} берутся из промежутка $[-\pi; \pi]$:

$$Q = \prod_{1 \le i < j \le n} R_{ij}(\varphi_{ij}). \qquad (10)$$

2-метод. Произвольную ортогональную матрицу 3-го порядка можно выразить через углы Эйлера α, β, γ:

$$Q(\alpha, \beta, \gamma) =$$

$$= \begin{pmatrix} \cos\alpha\cos\gamma - \sin\alpha\cos\beta\sin\gamma & -\cos\alpha\sin\gamma - \sin\alpha\cos\beta\cos\gamma & \sin\alpha\sin\beta \\ \sin\alpha\cos\gamma - \cos\alpha\cos\beta\sin\gamma & -\sin\alpha\sin\gamma + \cos\alpha\cos\beta\cos\gamma & -\cos\alpha\sin\beta \\ \sin\beta\sin\gamma & \sin\beta\cos\gamma & \cos\beta \end{pmatrix}.$$

Для наших **учебных** целей достаточно генерировать ортогональные матрицы небольшого размера (не более 6-го), и эти матрицы должны получаться нормированием по возможности целочисленных матриц с ненулевыми попарно ортогональными строками (или столбцами). В таком случае косинусы углов надо выбирать не сложными, например, $\frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{3}, \frac{2}{3}, \frac{3}{5}$.

3-й метод. Ортогональные матрицы можно получить, применяя процесс ортогонализации к строкам целочисленной квадратной матрицы A с ненулевым определителем (и последующей нормировкой всех строк). Процесс ортогонализации последовательно строит попарно ортогональные строки целочисленной матрицы B, в которой первая строка такая же, как и в матрице A. На k-м этапе из уже построенных попарно ортогональных вектор-строк $\mathbf{b}_1, \ldots, \mathbf{b}_k$, таких, что $L(\mathbf{a}_1, ..., \mathbf{a}_k) = L(\mathbf{b}_1, ..., \mathbf{b}_k)$ (L обозначает линейную оболочку), следующая вектор-строка $\mathbf{b}_{k+1}$ выбирается в виде $\mathbf{b}_{k+1} = \lambda_1\mathbf{b}_1 + ... + \lambda_k\mathbf{b}_k + \mu\mathbf{a}_{k+1}$ ($\lambda_i, \mu \in \mathbb{Z}$) из условия ортогональности всем векторам $\mathbf{b}_1, ..., \mathbf{b}_k$, откуда $\mu = -(\mathbf{b}_1)^2 ... (\mathbf{b}_k)^2$, $\lambda_i = -\dfrac{\mu(\mathbf{b}_k, \mathbf{a}_{k+1})}{(\mathbf{b}_i)^2}$. Числа $\lambda_1, ..., \lambda_k, \mu$ затем надо сократить на их наибольший общий делитель.

Проще всего в качестве матрицы A взять единичную матрицу, в которой в первой строке правее единицы стоят произвольные числа. Этот алгоритм почти детерминирован, произвольной является только эта первая строка. Тогда получается матрица B с треугольником нулей, например:

$$A = \begin{pmatrix} 1 & 1 & -2 & 1 & 3 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & -2 & 1 & 3 \\ -1 & 15 & -2 & -1 & -3 \\ 2 & 0 & 11 & 2 & 6 \\ -1 & 0 & 0 & 10 & -3 \\ -3 & 0 & 0 & 0 & 1 \end{pmatrix} = B.$$

Лучше взять случайную верхнетреугольную матрицу A с ненулевыми диагональными элементами (скажем, единицами), например:

$$A = \begin{pmatrix} 1 & 1 & 1 & -1 & -1 \\ 0 & 1 & 2 & -2 & -1 \\ 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 1 & -1 & -1 \\ -6 & -1 & 4 & -4 & 1 \\ 0 & -1 & 0 & 0 & -1 \\ -5 & 5 & 1 & 6 & -5 \\ 1 & -1 & 3 & 2 & 1 \end{pmatrix} = B.$$

4-й метод (пока не реализованный). Взять случайную целочисленную n-мерную строку с небольшими компонентами, например, $\mathbf{a}_1 = (1; -1; 0; 2)$, и в ее ортогональном дополнении $\mathbf{a}_1^{\perp}$, задаваемом уравнением $(\mathbf{a}_1, \mathbf{x}) = 0$, перебирая один за другим его целочисленные векторы, выбрать из первых N векторов вектор $\mathbf{a}_2$ с минимальной нормой, вычисляемой по формуле $\|\mathbf{b}\| = \max\limits_{1 \le i \le n} |b_i|$. Затем в ортогональном

дополнении векторов $\{\mathbf{a}_1, \mathbf{a}_2\}^{\perp}$, задаваемом матричным уравнением $A_2 X = 0$, где матрица A_2 состоит из двух строк $\mathbf{a}_1$ и $\mathbf{a}_2$, выбрать целочисленный вектор $\mathbf{a}_3$ с минимальной нормой (точнее, перебрать N векторов и выбрать из них минимальный по этой норме) и т. д. Этот способ требует наличия хорошо отлаженного генератора целочисленных векторов, ортогональных данным линейно независимым векторам $\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_k$, и поэтому его реализация пока затруднительна.

Заключение. На основании предложенных алгоритмов авторами создан и успешно применяется работающий прототип программного комплекса, позволяющий в автоматическом режиме составлять варианты контрольных работ и домашних заданий по аналитической геометрии и линейной алгебре.

ЛИТЕРАТУРА

[1] Карнаухов В.М., Русаков А.А. Компьютерный способ подготовки раздаточного материала контрольных работ по математике. *Материалы Междунар. науч.-прак. конф. «Информатизация образования — 2012»*. Орел, ООО «Картуш», 2012, 368 с.

[2] Карнаухов В.М. Приложение LATEX. Генератор вариантов контрольных работ. *Saarbrücken*, Germany, Lap lambert Academic Publishing, 2012. URL: http://karnauhov-60.narod.ru/

[3] Кручинин В.В., Морозова Ю.В. *Модели и алгоритмы генерации задач в компьютерном тестировании.* URL: http://www.lib.tpu.ru/fulltext/v/Bulletin_TPU/2004/v307/i5/29.pdf

[4] Amruth N. Kumar. Generation of Problems, Answers, Grade, and Feedback — Case Study of a Fully Automated Tutor. *ACM Journal of Educational Resources in Computing*, 2005, vol. 5(3).

[5] Гельфанд И.М. *Лекции по линейной алгебре*. Москва, Добросвет-2000, МЦНМО, 2007, 320 с.

[6] Курош А.Г. *Курс высшей алгебры*. Москва, Лань; Физматкнига, 2007, 432 с.

[7] Ильин В.А., Позняк Э.Г. *Линейная алгебра*. Москва, ФИЗМАТЛИТ, 2010, 280 с.

[8]. Гантмахер Ф.Р. *Теория матриц*. Москва, ФИЗМАТЛИТ, 2010, 560 с.

[9]. Соболев С.К., ред. *Сборник задач по линейной алгебре*. Москва, Изд-во МГТУ им. Н.Э. Баумана, 1991, 145 с.

[10] Соболев С.К. Генерирование матриц специального вида: аналитический подход. *Инженерный журнал: наука и инновация,* 2013, вып. 5.

Статья поступила в редакцию 28.06.2013

Ссылку на эту статью просим оформлять следующим образом:

Коновалов Я.Ю., Соболев С.К., Ермолаева М.А. Методические аспекты автоматической генерации задач по линейной алгебре. *Инженерный журнал: наука и инновации*, 2013, вып. 5. URL: http://engjournal.ru/catalog/pedagogika/hidden/740.html

Коновалов Ярослав Юрьевич окончил механико-математический факультет МГУ им. М.В. Ломоносова в 2005 г. и аспирантуру МГТУ им. Н.Э. Баумана

в 2009 г, ст. преподаватель кафедры «Высшая математика» МГТУ им. Н.Э. Баумана. Сфера научных интересов: атомарные функции, вейвлеты, цифровая обработка сигналов, численные методы. e-mail: kon20002000@mail.ru

Соболев Сергей Константинович окончил механико-математический факультет МГУ им. М.В. Ломоносова в 1973 г. и аспирантуру в МИАН им. В.А. Стеклова в 1976 г. Канд. физ.-мат. наук, доц. кафедры «Высшая математика» МГТУ им. Н.Э. Баумана. Автор ряда статей по математической логике, методике преподавания математики, методических пособий для студентов и старших школьников. Председатель методической комиссии кафедры «Высшая математика». e-mail: sergesobolev@mail.ru

Ермолаева Мария Андреевна окончила механико-математический факультет МГУ им. М.В. Ломоносова в 2002 г. и аспирантуру МГТУ им. Н.Э. Баумана в 2010 г. Ассистент кафедры «Высшая математика» МГТУ им. Н.Э. Баумана. Сфера научных интересов: непараметрические методы статистики. e-mail: mashik_@mail.ru