

О РЕКУРРЕНТНО-МАТРИЧНОЙ ФОРМЕ МЕТОДА ОПТИМАЛЬНОГО ИСКЛЮЧЕНИЯ

© 2009 А. И. Жданов, Л. В. Яблокова

Самарский государственный аэрокосмический университет

Рассматривается прямой рекуррентный метод, который является альтернативной формулировкой прямого проекционного метода решения систем линейных алгебраических уравнений, и показывается его эквивалентность методу оптимального исключения. Это позволяет рассматривать последний как прямой рекуррентный метод. Данный подход дает возможность создавать оптимальные по требуемому объему оперативной памяти машинные алгоритмы решения систем линейных алгебраических уравнений со многими правыми частями, которые особенно эффективны в алгоритмах итерационного уточнения.

Прямой рекуррентный метод, прямой проекционный метод, метод оптимального исключения.

§ 1. Введение

В данной работе рассматривается задача решения систем линейных алгебраических уравнений (СЛАУ) с невырожденными матрицами, т.е. задача вычисления решения СЛАУ:

$$Ax = b, A \in \mathbf{R}^{n \times n}, b \in \mathbf{R}^n, \text{rank } A = n. \quad (1)$$

Прямые методы решения СЛАУ оцениваются по трем важнейшим качествам: числу выполняемых арифметических операций, требованию к объему оперативной памяти и максимальной точности, достижимой при их помощи.

Наилучшими по вышеперечисленным показателям методами решения невырожденных СЛАУ продолжают оставаться методы типа исключения Гаусса. Как известно [1] - [3], число операций этих методов оценивается величиной $2n^3/3 + O(n^2)$, где в число операций включаются как умножения/деления, так и сложения/вычитания.

Гауссовское исключение существует во многих вариантах, которые алгебраически тождественны. Из всего множества вариантов исключения Гаусса метод оптимального исключения [1], [3] требует для своей реализации минимального объема оперативной памяти. Точно такие же характеристики имеет метод окаймления [1], [3], хотя он и не относится, вообще говоря, к методам исключе-

ния, так как по своей математической структуре принадлежит к классу методов малоранговой модификации. Однако метод окаймления представляет собой лишь некоторое видоизменение вычислительной схемы метода оптимального исключения [1] и не дает никаких преимуществ. Поэтому в дальнейшем будет рассматриваться метод оптимального исключения. В этом методе объем требуемой оперативной памяти оценивается при четном n величиной - $n^2/4$ машинных слов, а при нечетном n данный показатель будет близок к этому.

Как известно [4], в методе оптимального исключения попеременно выполняются операции прямого и обратного хода метода Гаусса. Это позволяет не вводить в оперативное запоминающее устройство очередную строку СЛАУ до тех пор, пока не преобразованы предыдущие строки, т.е. используется последовательный ввод строк матрицы и правых частей системы. Таким образом, метод оптимального исключения имеет рекуррентную структуру.

В настоящей работе рассматривается прямой рекуррентный метод решения СЛАУ, основанный на использовании формулы малоранговой модификации обратной матрицы (формулы Шермана-Моррисона [5]). Показана эквивалентность этого метода методу оп-

тимального исключения, хотя по своей алгоритмической структуре полученный рекуррентный метод не является методом исключения. Этот метод относится к классу прямых проекционных методов [6] – [8].

Указанная эквивалентность рассматриваемого прямого рекуррентного метода, прямого проекционного метода [8] и метода оптимального исключения дает возможность исследовать последний как конечный рекуррентный процесс, а также представлять его в компактной рекуррентно-матричной форме. Это позволяет получить оптимальные по требуемому объему оперативной памяти машинные алгоритмы решения СЛАУ со многими правыми частями, которые особенно эффективны в алгоритмах итерационного уточнения.

§ 2. Прямой рекуррентный алгоритм

Рассматриваемый в статье прямой рекуррентный метод решения СЛАУ [9] основан на формуле Шермана – Моррисона [5].

Л е м м а. Пусть дана матрица

$$B = \begin{pmatrix} U_{kk} & U_{k,n-k} \\ O_{n-k,k} & O_{n-k,n-k} \end{pmatrix} \in \mathbf{R}^{n \times n},$$

где $U_{kk} \in \mathbf{R}^{k \times k}$, $k < n$ и $\text{rank} U_{kk} = k$, $U_{k,n-k} \in \mathbf{R}^{k \times (n-k)}$, $O_{n-k,k}$ и $O_{n-k,n-k}$ – нулевые матрицы соответствующих размерностей.

Тогда для всех $|d| < s_{\min}(U_{kk})$ матрица

$$P_B = \lim_{d \rightarrow 0} (B + dE_n)^{-1} d$$

существует и равна

$$P_B = \begin{pmatrix} O_{kk} & -U_{kk}^{-1}U_{k,n-k} \\ O_{n-k,k} & E_{n-k} \end{pmatrix}, \quad (2)$$

где $d_{\min} U_{kk}$ – минимальное сингулярное число матрицы U_{kk} , E_n – единичная матрица порядка n , $k = 1, 2, \dots, n-1$.

Введем обозначения.

Пусть $i = 1, 2, \mathbf{K}, n$ – номера строк матрицы A , $a_i^T = (a_{i1}, \dots, a_{in})$ – строки матрицы A , τ – знак транспонирования,

$$A_i = \begin{pmatrix} a_1^{\circ} \\ \mathbb{M} \\ a_i^{\circ} \end{pmatrix} \in \mathbf{R}^{i \times n}, \quad A_n = A,$$

b_i – элементы вектора b ,

$$b = (b_1, \dots, b_n)^{\circ} \in \mathbf{R}^n,$$

$$\tilde{A}_i = \begin{pmatrix} a_{11} & \mathbf{L} & a_{1i} \\ \cdot & \cdot & \cdot \\ a_{i1} & \mathbf{L} & a_{ii} \end{pmatrix} \in \mathbf{R}^{i \times i}, \quad \tilde{A}_1 = a_{11},$$

$$\tilde{A}_n = A.$$

Теорема. Пусть все главные миноры матрицы A отличны от нуля, т.е.

$$\det \tilde{A}_i \neq 0 \quad \forall i = 1, 2, \mathbf{K}, n, \quad (3)$$

и пусть векторы $x_i \in \mathbf{R}^n$ и матрицы

$P_i \in \mathbf{R}^{n \times n}$, $i = 1, 2, \mathbf{K}, n$ удовлетворяют системе рекуррентных уравнений

$$x_{i+1} = x_i + g_{i+1}^i (b_{i+1} - a_{i+1}^{\circ} x_i) / w_{i+1}, \quad x_0 = 0, \quad (4)$$

$$P_{i+1} = P_i - g_{i+1}^i a_{i+1}^{\circ} P_i / w_{i+1}, \quad P_0 = E_n, \quad (5)$$

где $P_i = [g_1^i \mathbf{K} g_n^i] \in \mathbf{R}^{n \times n}$, $w_{i+1} = a_{i+1}^{\circ} g_{i+1}^i$, $i = 1, 2, \mathbf{K}, n-1$.

Тогда условия (2) являются необходимыми и достаточными для того, чтобы

$$w_{i+1} \neq 0 \quad \forall i = 1, 2, \mathbf{K}, n-1.$$

При этом $x_n = A^{-1}b$, т.е. x_n является решением СЛАУ(1).

Пусть $A_i = [\tilde{A}_i \quad A_{i,n-i}]$, где $\tilde{A}_i \in \mathbf{R}^{i \times i}$ и $A_{i,n-i} \in \mathbf{R}^{i \times (n-i)}$, тогда из условий (5) на ос-

новании леммы получаем, что $\forall i = 1, 2, \mathbf{K}, n$ матрицы $P_i = \lim_{d \rightarrow 0} P_i^d$ существуют и равны

$$P_i = \begin{pmatrix} O_{ii} & -\tilde{A}_i^{-1} A_{i,n-i} \\ O_{n-i,i} & E_{n-i} \end{pmatrix}. \quad (6)$$

Из (6) видно, что в матрице P_{i+1} первые $i+1$ векторов равны нулю, т.е. $g_j^{i+1} = 0$ $\forall j = 1, 2, \mathbf{K}, i-1$. Следовательно, формулу (5) можно записать в виде

$$g_j^{i+1} = g_j^i - \frac{y_j}{w_{i+1}} g_{i+1}^i, \quad (7)$$

где

$$y_j = a_{i+1}^T g_j^i, \quad j = i+2, \mathbf{K}, n, \quad i = 0, 1, \mathbf{K}, n-1,$$

$$\text{а } g_k^0 = e_k \quad \forall k = 1, 2, \mathbf{K}, n.$$

Из (7) непосредственно видно, что алгоритм (4), (5) является альтернативной формулировкой прямого проекционного метода [8].

Известно, что условия (2) не являются сильно ограничительными. Чтобы от них освободиться, достаточно выполнить перестановку, основанную на какой-либо стратегии выбора ведущих элементов. Как отмечается в [8], наиболее естественной и практичной для данного метода при решении СЛАУ с плотными матрицами продолжает оставаться стратегия с выбором максимального по модулю ведущего элемента по строке.

Формально эта стратегия на каждом i -м шаге состоит из следующих этапов:

1) определяется номер k из условия:

$$k = \min N_i, \text{ где}$$

$$N_i = \{k' : |a_{i+1}^T g_{k'}^i| = \max_{i+1 \leq l \leq n} |a_{i+1}^T g_l^i|\};$$

2) выполняется перестановка векторов

$$g_{i+1}^i \text{ и } g_k^i.$$

В [8] отмечается, что вычислительные эксперименты свидетельствуют о том, что прямой проекционный метод с рассмотрен-

ной стратегией выбора ведущих элементов лишь незначительно уступает по точности методу исключения с использованием LU -разложения и частичным выбором (по строке) ведущих элементов.

Для СЛАУ с разреженными матрицами в [8] предложена новая стратегия выбора ведущих элементов. В этой стратегии номер k выбирается на i -м шаге из условия: $k = \min N'_i$, где

$$N'_i = \{k' : |a_{i+1}^\diamond g_{k'}^i| \geq c \cdot \max_{i+1 \leq l \leq n} |a_{i+1}^\diamond g_l^i|, \quad 0 < c \leq 1\}.$$

В [8] показано, что эта стратегия в случае СЛАУ с разреженными матрицами при $c < 1$ позволяет значительно уменьшить заполнение матриц в процессе вычислений при незначительном снижении точности алгоритмов вида (6), (7).

§ 3. Анализ вычислительной схемы рекуррентного алгоритма

Нетрудно показать, что общее число арифметических операций, требуемых для реализации алгоритма (4), (5), оценивается величиной $4n^3 + O(n^2)$. Однако, учитывая специальную структуру (6) матриц P_i , можно представить алгоритм (4), (5) в эквивалентном виде, для которого общее число арифметических операций оценивается величиной $2n^3/3 + O(n^2)$.

Для этого предварительно введем следующие обозначения:

$$P_i = \begin{pmatrix} O_{ii} & Q_i \\ O_{n-i,i} & E_{n-i} \end{pmatrix},$$

$$Q_i = (q_1^i \dots q_{n-i}^i) = -\tilde{A}_i^{-1} A_{i,n-i} \in \mathbf{R}^{i \times (n-i)},$$

$$q_k^i \in \mathbf{R}^i, \quad k = 1, 2, \dots, n-i,$$

$$Q_i = \begin{pmatrix} q_2^i & q_3^i & \dots & q_{n-i}^i \\ 0 & 0 & \dots & 0 \end{pmatrix} \in \mathbf{R}^{(i+1) \times (n-i-1)}$$

для всех $i = 1, 2, \dots, n - 2$.

Из определения матриц Q_i и $\mathring{Q}_i$ следует, что матрицы Q_{i+1} и $\mathring{Q}_i$ имеют одинаковые размерности, т.е.

$$Q_{i+1}, \mathring{Q}_i \in \mathbf{R}^{(i+1) \times (n-i-1)}$$

для всех $i = 1, 2, \dots, n - 2$.

Представим строки a_i° матрицы A в виде двух компонентов:

$$(a_i)_u \in \mathbf{R}^i \text{ и } (a_i)_l \in \mathbf{R}^{n-i},$$

$$a_i^\circ = [(a_i)_u^\circ, (a_i)_l^\circ],$$

а также введем вектор y_i , состоящий из первых i компонент вектора x_i ,

$$y_i = (x_1^i, \dots, x_i^i)^\circ \in \mathbf{R}^i, \text{ где } x_i = (x_1^i, \dots, x_n^i)^\circ$$

$$\forall i = 1, 2, \mathbf{K}, n.$$

Тогда, используя введенные обозначения, непосредственно получаем, что алгоритм (4), (5) можно записать в эквивалентном виде:

$$\mathring{Q}_1 = -(a_{12}, \dots, a_{1n}) / a_{11}, \quad (8)$$

$$Q_{i+1} = \mathring{Q}_i - \frac{z_i n_{i+1}^\circ}{\omega_{i+1}}, \quad (9)$$

$$i = 1, 2, \dots, n - 1, n \geq 2,$$

$$y_1 = b_1 / a_{11}, \quad (10)$$

$$y_{i+1} = \begin{pmatrix} y_i \\ 0 \end{pmatrix} + \frac{z_i (b_{i+1} - \mathring{a}_{i+1}^\circ y_i)}{\omega_{i+1}}, \quad (11)$$

$$i = 1, 2, \dots, n - 1, n \geq 2,$$

где

$$z_i = \begin{pmatrix} q_1^i \\ 1 \end{pmatrix} \in \mathbf{R}^{i+1},$$

$$\mathring{a}_{i+1} = (a_{i+1,1}, \dots, a_{i+1,i})^\circ \in \mathbf{R}^i,$$

$$n_{i+1}^\circ = (a_{i+1})_u^\circ \mathring{Q}_i + (a_{i+1})_l^\circ,$$

$$\omega_{i+1} = (a_{i+1})_u^\circ z_i.$$

Таким образом, если выполняются условия теоремы, то $y_n = x_n$.

Рекуррентная формула (5) эквивалентна рекуррентным формулам (8)-(9), а рекуррентная формула (4) - формулам (10)-(11).

Из формул (8)-(11) непосредственно видно, что вычислительная схема предлагаемого алгоритма практически совпадает с вычислительной схемой метода оптимального исключения [3]. Следовательно, число его арифметических операций оценивается величиной $2n^3/3 + O(n^2)$, а для решения СЛАУ n -го порядка этим методом достаточно иметь оперативную память машины порядка $n^2/4$ слов. Необходимая для машинной реализации алгоритма (8)-(11) оперативная память определяется максимальным размером матрицы Q_i .

Таким образом, формулы (4), (5) (или, что эквивалентно, формулы (8) — (11)) представляют рекуррентно-матричную форму метода оптимального исключения.

§ 4. Решение систем со многими правыми частями

В [3] отмечалось, что единственным серьезным недостатком метода оптимального исключения по сравнению с LU -разложением является то, что он не позволяет эффективно решать системы со многими правыми частями, так как для его реализации требуется достаточно большой объем оперативной памяти, позволяющий запомнить все преобразования с матрицей системы. Использо-

ние рекуррентно - матричной формы алгоритма (формул (8), (9) и (10), (11)) дает возможность в значительной мере устранить этот недостаток классической вычислительной схемы метода оптимального исключения.

Подробно рассмотрим лишь общий случай СЛАУ с плотными матрицами.

Формулы (8), (9) и (10), (11) позволяют полностью разделить вычисления в методе оптимального исключения на два этапа аналогично использованию LU -разложения в методе исключения Гаусса. Первый этап — это обработка элементов матрицы A по формулам (8), (9). Второй этап — обработка правых частей и вычисление решений по формулам (10), (11). Точно так же, как и при LU -разложении, основная масса вычислений (примерно $2n^3/3 + O(n^2)$) приходится здесь на вычисления по формулам (8), (9). Из формул (10), (11) видно, что для выполнения второго этапа нет необходимости запоминать на первом этапе все матрицы Q_i . Достаточно запомнить лишь векторы $q_1^1, \dots, q_1^{n-1}$ и числа $w_1, \dots, w_n$, где $w_1 = a_{11}$. Они могут быть размещены в верхней треугольной матрице

$$R = (r_1 \dots r_n) \in \mathbf{R}^{n \times n},$$

где

$$r_1 = (w_1, 0, \dots, 0)^{\circ},$$

$$r_2 = \left((q_1^1)^{\circ}, w_2, 0, \dots, 0 \right)^{\circ}, \dots,$$

$$r_{n-1} = \left((q_1^{n-2})^{\circ}, w_{n-1}, 0 \right)^{\circ},$$

$$r_n = \left((q_1^{n-1})^{\circ}, w_n \right)^{\circ} \in \mathbf{R}^n.$$

Таким образом, для их хранения требуется оперативная память объемом $n(n+1)/2$.

В этом же массиве в процессе вычислений на первом этапе (по формулам (8), (9)) может храниться и текущая матрица Q_i .

Для реализации второго этапа (вычислений по формулам (10), (11)) кроме верхней треугольной матрицы R требуются элементы исходной матрицы A , расположенные ниже главной диагонали, т.е. элементы матрицы

$$L_A = (l_{ij}), \text{ где}$$

$$l_{ij} = \begin{cases} a_{ij}, & \text{если } i > j, \\ 0, & \text{если } i \leq j. \end{cases}$$

Для хранения элементов матрицы L_A требуется $n(n-1)/2$ машинных слов.

При этом возможны два варианта. В первом варианте элементы матрицы L_A хранятся в оперативной памяти, т.е. при обработке элементов матрицы A на первом этапе происходит запоминание в оперативной памяти части элементов матрицы A в матрице L_A . Во втором варианте на первом этапе не происходит запоминания в оперативной памяти элементов матрицы L_A и они сохраняются лишь на внешнем запоминающем устройстве.

В первом варианте для реализации рекуррентно - матричной формы метода оптимального исключения требуется хранить в оперативной памяти машины элементы матрицы $F = L_A + R$, т.е. требуется оперативная память объемом n^2 машинных слов. Следовательно, в этом варианте рекуррентно - матричная форма алгоритма не имеет преимуществ перед классической вычислительной схемой, основанной на LU -разложении матрицы A .

Во втором варианте требуется хранить в оперативной памяти лишь матрицу R и, следовательно, требуемый объем оперативной памяти — $n(n+1)/2$ машинных слов. Таким образом, в этом варианте рекуррентно - мат-

ричная форма метода оптимального исключения дает выигрыш в требуемом объеме оперативной памяти в $2n/(n+1)$ -раз, т.е. приблизительно в два раза по сравнению с вычислительной схемой, основанной на LU -разложении матрицы A . Однако в этом варианте требуется постоянный обмен с внешней памятью (для чтения элементов матрицы L_A), что снижает быстроедействие этого метода по сравнению с аналогичным вариантом, основанным на LU -разложении.

Известно [2], что при итерационном уточнении возникает необходимость решения СЛАУ с несколькими правыми частями. При этом, если в итерационном уточнении используется LU -разложение, то предъявляются следующие требования к объему оперативной памяти машин: $2n^2$ машинных слов при хранении матрицы A в оперативной памяти и n^2 при хранении матрицы A во внешней памяти. Это связано с тем, что для итерационного уточнения кроме LU -разложения требуется на каждой итерации еще и исходная матрица A .

Рассмотрим, как изменятся соответствующие характеристики, если в итерационном уточнении вместо LU -разложения использовать рекуррентно - матричную форму метода оптимального исключения, т.е. формулы (8)–(9). Если матрица A хранится в оперативной памяти, то отпадает необходимость в матрице L_A , и, следовательно, достаточно хранить только матрицы A и R .

Таким образом, требуемый в этом случае объем оперативной памяти равен $n^2 + n(n+1)/2 = 3n^2/2 + n/2$ машинных слов.

Если матрица A хранится во внешней памяти, то соответственно для итерационного уточнения требуется объем оперативной памяти, равный $n^2/2 + n/2$ машинных слов.

Следовательно, использование в итерационном уточнении рекуррентно-матричной формы метода оптимального исключения предпочтительнее LU -разложения с точки зрения требуемой оперативной памяти при любом способе хранения исходной матрицы A . При этом, как отмечалось в § 2, почти оди-

наково ведут себя оба метода с точки зрения устойчивости к ошибкам округления (при соответствующих стратегиях выбора ведущих элементов).

Очевидно, что в случае СЛАУ с разреженными матрицами данный алгоритм будет еще более эффективен по сравнению с LU -разложением с точки зрения требуемой оперативной памяти.

Библиографический список

1. Воеводин, В. В. Численные методы алгебры (теория и алгоритмы) [Текст] / В. В. Воеводин. - М.: Наука, 1966. - 248с.
2. Воеводин, В. В. Вычислительные основы линейной алгебры [Текст] / В. В. Воеводин. - М.: Наука, 1977. - 303с.
3. Воеводин, В. В. Матрицы и вычисления [Текст] / В. В. Воеводин, Ю. А. Кузнецов. - М.: Наука, 1984. - 318 с.
4. Беклемишев, Д.В. Дополнительные главы линейной алгебры [Текст] / Д. В. Беклемишев. - М.: Наука, 1983. - 336 с.
5. Ортега, Дж. Итерационные методы решения нелинейных систем уравнений со многими неизвестными [Текст] / Дж. Ортега, В. Рейнболдт. - М.: Мир, 1975. - 558 с.
6. Фаддеев, Д. К. Вычислительные основы линейной алгебры [Текст] / Д. К. Фаддеев, В. Н. Фаддеева. - М.: Физматгиз, 1960. - 656 с.
7. Abaffy J., Broyden C., Spedicato E. A class of direct methods for linear equations [Текст] / J. Abaffy, C. Broyden, E. Spedicato // Numer. Math. 1984. V.45. P.361–376.
8. Benzi M., Meyer C.D. A direct projection method for sparse linear systems [Текст] / M. Benzi, C. D. Meyer // SIAM J. Sci. Comput. 1995. V.16. N.5. P.1159–1176.
9. Жданов, А. И. Прямой последовательный метод решения систем линейных уравнений [Текст] / А. И. Жданов // Докл. РАН. - 1997. - Т.356, № 5. - С. 442 – 444.
10. Малышев, А. Н. Введение в вычислительную линейную алгебру [Текст] / А. Н. Малышев. - Новосибирск: Наука, 1991. - 228 с.
11. Гантмахер, Ф. Р. Теория матриц [Текст] / Ф. Р. Гантмахер. - М.: Наука, 1966. - 576 с.

References

1. Voyevodin, V. V. Numerical methods in algebra (theory and algorithms) / V. V. Voyevodin – Moscow: Nauka, 1966 – 248 pp.
2. Voyevodin, V. V. Computing foundations of linear algebra / V. V. Voyevodin – Moscow: Nauka, 1977 – 303 pđ.
3. Voyevodin, V. V., Kuznetsov Yu. A. Matrices and computations / V. V. Voyevodin, Yu. A. Kuznetsov – Moscow: Nauka, 1984 – 318 pp.
4. Beklemishev, D. V. Additional chapters of linear algebra / D. V. Beklemishev – Moscow: Nauka, 1983 – 336 pp.
5. Ortega, J. Iteration methods of solving non-linear systems of equations with many unknowns / J. Ortega, V. Reinboldt – Moscow: Mir, 1975 – 558 pp.
6. Faddeyev, D. K. Computing foundations of linear algebra / D. K. Faddeyev, V. N. Faddeyeva – Moscow: Fizmatgiz, 1960 – 656 pp.
7. Abaffy J., Broyden C., Spedicato E. A class of direct methods for linear equations / J. Abaffy, C. Broyden, E. Spedicato // Numer. Math. 1984. V.45. P.361–376.
8. Benzi M., Meyer C. D. A direct projection method for sparse linear systems / Ĭ. Benzi, C. D. Meyer // SIAM J. Sci. Comput. 1995. V.16. N.5. Đ.1159–1176.
9. Zhdanov, A. I. Direct sequential method of solving linear equation systems / A. I. Zhdanov // Report to the Russian Academy of Sciences. – 1997 – vol. 356, No. 5 – pp.442 – 444.
10. Malyshev, A. N. Introduction to computing linear algebra / A. N. Malyshev – Novosibirsk: Nauka, 1991 – 228 pp.
11. Gantmakher F. R. Theory of matrices / F. R. Gantmakher – Moscow: Nauka, 1966 – 576 pp.

RECURRENT MATRIX FORM OF THE OPTIMAL ELIMINATION METHOD

© 2009 A. I. Zhdanov, L. V. Yablokova

Samara State Aerospace University

The paper deals with the direct recurrent method [1] which is an alternative formulation of the direct projection method of solving systems of linear algebraic equations. The direct recurrent method is shown to be equivalent to the method of optimal elimination, which makes it possible to treat the latter as the direct recurrent method. This approach provides the potential for creating machine algorithms of solving systems of linear algebraic equations with many right sides, especially efficient in iteration refinement algorithms. The algorithms created have an optimal volume of on-line storage.

Direct recurrent method, direct projection method, method of optimal elimination.

Информация об авторах

Жданов Александр Иванович, заведующий кафедрой прикладной математики, доктор физико-математических наук, профессор, Самарский государственный аэрокосмический университет; e-mail: zhdanov@ssau.ru. Область научных интересов: вычислительная математика, матричные вычисления.

Яблокова Людмила Вениаминовна, ассистент кафедры прикладной математики, Самарский государственный аэрокосмический университет; e-mail: lyablokova@gml.com Область научных интересов: вычислительная математика.

Zhdanov Alexander Ivanovitch, Samara State Aerospace University, head of the department of applied mathematics, doctor of physical and mathematical science, professor, e-mail: zhdanov@ssau.ru. Area of research: computing mathematics, matrix computations.

Yablokova Lyudmila Veniaminovna, Samara State Aerospace University, assistant of the department of applied mathematics, e-mail: lyablokova@gml.com. Area of research: computing mathematics.