

**М. М. Гуарий**, канд. техн. наук, e-mail: gourary@ippm.ru,  
**С. Г. Русаков**, д-р техн. наук, проф., e-mail: rusakov@ippm.ru,  
**С. Л. Ульянов**, д-р техн. наук, e-mail: ulyas@ippm.ru,  
**М. М. Жаров**, канд. техн. наук, e-mail: zarov@ippm.ru,

ФГБУН Институт проблем проектирования в микроэлектронике РАН, Москва

## Итерационный метод решения параметризованной системы линейных алгебраических уравнений с многими правыми частями

*Рассматривается задача решения систем линейных алгебраических уравнений со многими правыми частями и матрицей, зависящей от параметра. Предлагается итерационный метод решения с повторным использованием результатов матрично-векторного умножения, который обеспечивает сокращение вычислительных затрат при решении таких систем. Метод обобщает ряд известных алгоритмов на базе подпространства Крылова. Применение метода показано на примере выполнения периодического малосигнального анализа нелинейных электронных схем.*

**Ключевые слова:** системы линейных алгебраических уравнений, линейные преобразования, ортогонализация, итерационные методы, подпространство Крылова, приближенное решение

### Введение

В ряде задач из различных технических приложений возникает необходимость многократного решения систем линейных алгебраических уравнений (СЛАУ) с различными правыми частями. В данной работе рассматривается задача решения высокоразмерной параметризованной СЛАУ вида

$$A(s_m)x^{(m)} = b^{(m)}, \quad m = 1, \dots, M, \quad (1)$$

где матрица системы линейно зависит от некоторого скалярного параметра  $s$ :

$$A(s) = A' + sA''. \quad (2)$$

Такая задача возникает, например, при расчетах малосигнальных частотных характеристик нелинейных электронных схем ( $s$  — частота) или при решении задач гомотопии, где  $s$  — параметр продолжения. Предполагается, что система (1) имеет высокую размерность.

Решение линейной системы (1) может быть выполнено с помощью как прямых, так и итерационных методов, однако итерационные методы более предпочтительны вследствие высокой размерности системы. Применение итерацион-

ных методов для решения линейных систем с постоянными матрицами высокой размерности позволяет значительно ускорить решение системы и в ряде случаев не имеет альтернативы [1]. Наибольшее распространение получили методы подпространства Крылова QMR [2] и GMRES [3], причем метод GMRES оказался более эффективным при решении практических задач [4]. Однако непосредственное применение метода GMRES для решения линейной системы (1) с многими правыми частями и матрицей (2) становится неэффективным вследствие линейного роста вычислительных затрат с увеличением числа правых частей. Рост затрат обусловлен тем, что в методе используется базис подпространства Крылова, который жестко связан с вектором начальной невязки и значениями элементов матрицы. Поэтому базис, полученный для одного значения параметра  $s$ , не может быть использован для другого значения ввиду изменения элементов матрицы.

Для решения линейных систем с многими правыми частями разработаны несколько итерационных методов [5–9]. В работах [10–13] предложены методы решения линейной системы вида (1) для случая, когда матрица  $A''$  равна

единичной матрице. Для построения итерационного метода в этом случае используется свойство инвариантности подпространства Крылова  $\mathbf{K}_k(A + sI, b) = \mathbf{K}_k(A, b)$ , которое позволяет использовать единый базис подпространства для решения систем с различными значениями параметра  $s$ . Если матрица  $A''$  отлична от единичной, то необходимый вид линейной системы можно было бы получить, умножая ее на обратную к  $A''$  матрицу (предобуславливание). Этот подход требует явного формирования и вычисления обратной матрицы. Так как основной операцией в методах рассматриваемого класса является матрично-векторное умножение, которое может выполняться без явного формирования матриц, то получение обратной матрицы практически исключено. Поэтому актуальной задачей является конструирование алгоритмов решения параметризованных линейных систем вида (1) с произвольными несимметричными матрицами  $A', A''$ .

В работе предлагается подход к последовательному решению линейных систем (1), при котором каждое последующее решение определяется с помощью минимизации невязки при использовании результатов матрично-векторного умножения, вычисленных при получении предыдущих решений. При этом достигается сокращение вычислительных затрат за счет уменьшения операций матрично-векторного умножения и повторного использования найденных векторов. Полученные таким образом базисные векторы определяют приближенное решение линейной системы, но не принадлежат подпространству Крылова. Поэтому вначале (разд. 1) рассмотрены методы минимальной невязки решения СЛАУ, которые допускают генерацию векторов с помощью операций матрично-векторного умножения. Эти методы являются обобщением известных алгоритмов на базе подпространств Крылова. В разд. 2 показано, что дополнительное ускорение предложенных методов минимальной невязки может быть достигнуто за счет порядка и способов применения линейных преобразований. Затем в разд. 3 рассмотрены методы для решения параметризованной линейной системы с многими правыми частями. Представлен алгоритм с использованием ортогонализации Грама — Шмидта и алгоритм без ортогонализации, в котором формирование и разложение матрицы Грама выполняются с учетом элементов матрицы, полученных при решении предыдущих систем. Применение и характеристики предложенных алгоритмов рассматриваются в разд. 4.

## 1. Алгоритм минимальной невязки

Рассматривается приближенное решение линейной системы

$$Ax = b, \quad (3)$$

где  $A$  — квадратная невырожденная матрица.

Пусть имеется вычислительная процедура  $\text{GET}(u_k, v_k)$ , которая генерирует последовательность пар векторов  $u_k, v_k$  таких, что

$$v_k = Au_k, \quad k = 1, 2, \dots \quad (4)$$

Векторы  $u_k$  назовем исходными, а векторы  $v_k$  — результирующими.

Тогда под алгоритмом минимальной невязки на основе матрично-векторного произведения (МНМВП) понимается алгоритм, который находит приближенное решение линейной системы (3) в виде

$$x_K = Uc = \sum_{k=1}^K c_k u_k. \quad (5)$$

Здесь  $U$  — матрица, столбцами которой служат векторы  $u_k$ ;  $c$  — вектор-столбец коэффициентов  $c_k$ , которые находятся из условия минимума нормы невязки линейной системы (3):

$$\|Ax_K - b\|_2 \rightarrow \min. \quad (6)$$

Принимая во внимание (4—6), можно записать

$$\|r_K\| = \left\| \sum_{k=1}^K c_k v_k - b \right\| \rightarrow \min. \quad (7)$$

Таким образом, для определения решения в виде (5) необходимо решить задачу минимизации (7) относительно коэффициентов  $c_k$ . Задача может быть решена с помощью формирования ортонормированного базиса в пространстве результирующих векторов  $\hat{v}_1, \dots, \hat{v}_K$  и соответствующих им исходных векторов  $\hat{u}_k$ ,  $\hat{v}_k = A\hat{u}_k$ . Тогда коэффициенты приближенного решения

$$x_K = \hat{U}\hat{c} = \sum_{k=1}^K \hat{c}_k \hat{u}_k \quad (8)$$

определяются с помощью проекции результирующих векторов на вектор  $b$ :

$$\hat{c}_k = (\hat{v}_k, b). \quad (9)$$

Векторы  $\hat{v}_k$  могут быть получены с помощью процесса ортогонализации Грама — Шмидта [1]:

$$\hat{v}_k = v_k - \sum_{i=1}^k (v_k, \hat{v}_i) \hat{v}_i, \quad \hat{v}_k = \hat{v}_k / \|\hat{v}_k\|. \quad (10)$$

Исходные векторы определяются аналогично:

$$\hat{u}_k = u_k - \sum_{i=1}^k (v_k, \hat{v}_i) \hat{u}_i, \hat{u}_k = \hat{u}_k / \|v_k\|. \quad (11)$$

Справедливы также рекуррентные соотношения

$$r_k = r_{k-1} - \hat{c}_k \hat{v}_k, r_0 = b; \quad (12)$$

$$x_k = x_{k-1} + \hat{c}_k \hat{u}_k, x_0 = 0. \quad (13)$$

Процесс вычислений по формулам (9)—(13) может быть записан в виде следующего алгоритма МНМВП.

*Шаг 1.* Инициализация  $r = b, x = 0$

*Шаг 2.* Цикл  $k = 1, 2, \dots$

Шаг 2.1. Get  $(u_k, v_k)$

Шаг 2.2. Цикл  $j = 1, \dots, k - 1$

$$v_k = v_k - (v_j, v_k) v_j, u_k = u_k - (v_j, v_k) u_j$$

Шаг 2.3.  $v_k = v_k / \|v_k\|, u_k = u_k / \|v_k\|,$

Шаг 2.4.  $r = r - (v_k, r) v_k, x = x + (v_k, r) u_k$

В процедуре GET( $u_k, v_k$ ) выполняется назначение исходного вектора  $u_k$  и вычисление результирующего вектора  $v_k$ . В общем случае исходный вектор  $u_k$  рассчитывается по формуле

$$u_k = P^{-1}(\alpha_k \hat{v}_{k-1} + \beta_k r_{k-1}), v_k = Au_k, \quad (14)$$

где  $\alpha_k, \beta_k$  — коэффициенты и  $\alpha_1 = 0, P$  — матрица правого преобуславливателя.

В зависимости от выбора коэффициентов исходного вектора  $u_k$  в процедуре Get( $u_k, v_k$ ) можно получить те или иные известные алгоритмы. Например, выбор  $\alpha_k = 0, \beta_k = 1$  соответствует алгоритму GCR [1], а  $\alpha_k = 1, \beta_k = 0$  дает

$$u_k = P_k^{-1} \hat{v}_{k-1}, v_k = Au_k, \quad (15)$$

что приводит к алгоритму ORTHODIR [1].

## 2. Сокращение вычислительных затрат в алгоритме МНМВП

1. **Алгоритм МНМВП-Т.** Линейное преобразование исходных векторов (11) требует значительных вычислительных затрат. Этих затрат можно избежать, если применить это линейное преобразование не к исходным векторам, а к вектору коэффициентов  $\hat{c}$  (9) для получения вектора  $c$ . Для получения необходимого преобразования отметим, что процесс ортого-

нализации (10) может быть записан в форме QR разложения матрицы результирующих векторов  $V$ :

$$V = \hat{V}\Gamma, \quad (16)$$

где  $\Gamma$  — верхняя треугольная матрица с элементами  $\gamma_{ij} = (v_j, \hat{v}_i)$ .

Аналогично для матрицы исходных векторов

$$U = \hat{U}\Gamma. \quad (17)$$

Поэтому вектор решения (5) имеет вид

$$x_K = Uc = \hat{U}\Gamma c = \hat{U}(\Gamma c), \quad (18)$$

откуда с учетом (8) получаем

$$c = \Gamma^{-1} \hat{c}. \quad (19)$$

Тогда алгоритм МНМВП можно переписать, включив формирование и решение системы с верхней треугольной матрицей.

**Алгоритм МНМВП-Г**

*Шаг 1.* Инициализация  $r = b, x = 0$

*Шаг 2.* Цикл  $k = 1, 2, \dots$

Шаг 2.1. Get  $(u_k, v_k)$

Шаг 2.2. Цикл  $j = 1, \dots, k - 1$

$$\gamma_{jk} = (v_j, v_k), v_k = v_k - \gamma_{jk} v_j$$

Шаг 2.3.  $\gamma_{kk} = \|v_k\|, v_k = v_k / \gamma_{kk}$

Шаг 2.4.  $c_k = (v_k, r), r = r - c_k v_k$

*Шаг 3.*  $c = \Gamma^{-1} c, x = \sum_{i=1}^k c_i u_i$

Здесь вектор  $c$  содержит элементы вектора  $\hat{c}$  и на последнем шаге элементы вектора  $c$ .

2. **Алгоритм МНМВП-Г.** Представим норму невязки (7) в виде квадратной функции аргумента  $c_k$ :

$$\|r_K\|^2 = c^T G c - 2p^T c + \|b\|^2 \rightarrow \min. \quad (20)$$

Здесь  $G = V^T V$  — матрица Грама с элементами  $g_{ij} = (v_i, v_j), p$  — вектор с компонентами  $p_i = (b, v_i)$ .

Решение линейной системы

$$Gc = p \quad (21)$$

доставляет минимум выражения (20).

Поскольку матрица Грама симметричная и положительно определенная, то для решения системы (21) целесообразно использовать разложение Холецкого, т.е. представление матрицы в виде произведения транспонированной верхней треугольной матрицы на верхнюю треугольную матрицу.

Для матрицы Грама верхняя треугольная матрица разложения Холецкого совпадает с верхней треугольной матрицей  $\Gamma$  (16):

$$G = V^T V = (\hat{V}\Gamma)^T (\hat{V}\Gamma) = \Gamma^T (\hat{V}^T \hat{V}) \Gamma = \Gamma^T \Gamma. \quad (22)$$

Поэтому для определения коэффициентов решения необходимо решить две линейные системы с нижней и верхней треугольными матрицами:

$$\hat{c} = \Gamma^{-T} p; \quad (23)$$

$$c = \Gamma^{-1} \hat{c}. \quad (24)$$

Текущий вектор невязки в этом случае определяется по формуле

$$r_k = b - \sum_{i=1}^k c_k v_k \quad (25)$$

или по рекуррентному соотношению

$$r_k = r_{k-1} - \hat{c}_k \left( \sum_{i=1}^{k-1} \gamma_{ik} v_i / \gamma_{kk} + v_k \right). \quad (26)$$

Тогда алгоритм МНМВП, включающий разложение матрицы Грама и решение систем с треугольной матрицей (алгоритм МНМВП-Г), имеет следующий вид:

*Шаг 1.* Инициализация  $r = b$ ,  $x = 0$

*Шаг 2.* Цикл  $k = 1, 2, \dots$

Шаг 2.1. Get  $(u_k, v_k)$

Шаг 2.2. Цикл  $j = 1, \dots, k$

Если  $j < k$

$$\gamma_{jk} = \left( (v_j, v_k) - \sum_{i=1}^{j-1} \gamma_{ik} \gamma_{ij} \right) / \gamma_{jj}$$

Иначе  $\gamma_{kk} = \sqrt{(v_k, v_k) - \sum_{i=1}^{k-1} \gamma_{ik}^2}$

Шаг 2.3.  $c_k = \left( (b, v_k) - \sum_{i=1}^{k-1} \gamma_{ik} c_i \right) / \gamma_{kk}$

Шаг 2.4  $r = r - c_k \left( \sum_{i=1}^{k-1} \gamma_{ik} v_i / \gamma_{kk} + v_k \right)$

*Шаг 3.*  $c = \Gamma^{-1} c$ ,  $x = \sum_{i=1}^k c_i u_i$

На шаге 2.2 выполняется разложение Холецкого матрицы Грама. На шаге 2.3 находится решение линейной системы с нижней треугольной матрицей (23).

### 3. Алгоритмы решения параметризованной линейной системы

1. Алгоритм ММН с ортогонализацией. Нашей целью является построение такого алго-

ритма решения линейных систем (1), в котором для получения последующего решения используются результаты матрично-векторного умножения, вычисленные при получении предыдущих решений. Новый алгоритм получается на основе алгоритма МНМВП-Т.

Матрично-векторное произведение с матрицей (2) имеет вид

$$v(s) = A(s)u = v' + sv'', \quad (27)$$

где

$$v' = A'u; \quad v'' = A''u. \quad (28)$$

Из выражения (27) следует, что если вычислить векторы  $v'$ ,  $v''$  при некотором значении параметра  $s = s_1$ , то матрично-векторное произведение  $A(s)u$  легко может быть получено для любого другого значения параметра по формуле (27). Такая операция включает умножение вектора на скаляр и сложение двух векторов. При этом вычислительные затраты значительно меньше затрат на выполнение нового матрично-векторного умножения. Вычисленные векторы необходимо хранить в памяти программы. При добавлении нового вектора произведения для получения ортонормированного базиса выполняется ортогонализация. Так как найденные базисные векторы уже не принадлежат пространству Крылова, то новый вектор произведения может оказаться линейной комбинацией векторов текущего базиса. В этом случае происходит обрыв вычислительного процесса. Проверка условия

$$(v_k, r_k) \neq 0 \quad (29)$$

позволяет определить возможность обрыва.

При выполнении этого условия обрыв процесса исключается, так как текущий вектор невязки ортогонален базисным векторам. Если условие (29) не выполняется, возможно выполнение одного из следующих действий:

— пропуск вектора, если в условии (29) участвует вектор произведения из памяти программы;

— перевычисление вектора, полученного с помощью матрично-векторного умножения, по формуле

$$v_k = (AP^{-1})^m v_k, \quad (30)$$

где  $m$  — целое, при котором удовлетворяется условие (29).

Тогда процедура получения пары векторов Get  $(u_k, v_k)$  имеет вид

Шаг 1.  $l_k = l_k + 1$

Шаг 2. Если  $l_k > n$

Шаг 2.1.  $n = l_k$

Шаг 2.2. Если  $breakdown = NO$

$u_n = P^{-1}r$

иначе  $u_n = P^{-1}v_k$

Шаг 2.3.  $v'_n = A'u_n$ ;  $v''_n = A''u_n$

Шаг 3.  $v_k = v'_{l_k} + s_m v''_{l_k}$

Шаг 4. Если  $(v_k, r) = 0$

$breakdown = YES$ ;

Переход на Шаг 1.

Здесь  $n$  — число пар векторов  $u_k, v_k$ , хранимых в памяти программы;  $l_k$  — индекс пары векторов на шаге  $k$  алгоритма МНМВП-Т.

Отметим, что шаг инициализации алгоритма теперь имеет вид  $r = b^{(m)}$ ,  $x^{(m)} = 0$ , шаг вычисления приближенного решения —  $x^{(m)} = \sum_{i=1}^k c_{l_i} u_i$ . Остальные шаги алгоритма МНМВП-Т

не изменяются. Вариант алгоритма МНМВП-Т для решения параметризованных линейных систем назовем ММН алгоритмом.

## 2. Алгоритм ММН без ортогонализации.

ММН алгоритм без ортогонализации может быть построен на базе алгоритма МНМВП-Г, в котором исключаются вычислительные затраты на ортогонализацию векторов. Кроме того, можно снизить затраты на получение матрицы Грама.

Используя определение матрицы Грама, можно записать ее элементы:

$$g_{ij}(s) = (v_i(s), v_j(s)) = g'_{ij} + s g''_{ij} + s^2 g'''_{ij}, \quad (31)$$

где

$$g'_{ij} = (v'_i, v'_j); \quad g''_{ij} = (v'_i, v''_j) + (v''_i, v'_j); \quad g'''_{ij} = (v''_i, v''_j). \quad (32)$$

Из выражений (31, 32) следует, что, располагая значениями четырех скалярных произведений, можно вычислить элементы матрицы для любого параметра  $s$ . Кроме того, получив элементы матрицы для трех значений параметра  $s^{(1)}, s^{(2)}, s^{(3)}$ , можно использовать интерполяционную формулу Лангранжа [14]:

$$g_{ij}(s) = L(s, g_{ij}^{(1)}, g_{ij}^{(2)}, g_{ij}^{(3)}, s^{(1)}, s^{(2)}, s^{(3)}) = \\ = \frac{g_{ij}^{(1)}(s - s^{(2)})(s - s^{(3)})}{(s^{(1)} - s^{(2)})(s^{(1)} - s^{(3)})} + \frac{g_{ij}^{(2)}(s - s^{(1)})(s - s^{(3)})}{(s^{(2)} - s^{(1)})(s^{(2)} - s^{(3)})} + \\ + \frac{g_{ij}^{(3)}(s - s^{(1)})(s - s^{(2)})}{(s^{(3)} - s^{(1)})(s^{(3)} - s^{(2)})}, \quad (33)$$

где  $g_{ij}^{(l)} = g_{ij}(s^{(l)})$ .

Дополнительно можно исключить вычисление на каждой итерации вектора невязки по формуле (25), а для расчета нормы вектора невязки для проверки сходимости алгоритма использовать следующее выражение:

$$\|r_k\|^2 = \|r_{k-1}\|^2 - \hat{c}_k^2. \quad (34)$$

Ниже представлен алгоритм, использующий формулы (33), (34) для решения системы линейных уравнений (1) с параметром  $s_m$ .

### Алгоритм ММН-Г.

Шаг 1. Инициализация  $r = b^{(m)}$ ,  $\rho = \|r\|$ ,  $x = 0$

Шаг 2. Цикл  $k = 1, 2, \dots$

Шаг 2.1. Если  $k > n$

Шаг 2.1.1.  $n = k$ ;  $l_k = 0$

Шаг 2.1.2.  $u_k = P^{-1}r$ ;  $v'_k = A'u_k$ ;  $v''_k = A''u_k$

Шаг 2.2.  $v_k = v'_k + s_m v''_k$

Шаг 2.3.  $l_k = l_k + 1$ ,  $l = l_k$

Шаг 2.4. Если  $l < 3$

$s_k^{(l)} = s_m$

Шаг 2.5. Цикл  $j = 1, \dots, k$

Шаг 2.5.1.

Если  $(l > 3)$

$g = L(s_m, g_{kj}^{(1)}, g_{kj}^{(2)}, g_{kj}^{(3)}, s_k^{(1)}, s_k^{(2)}, s_k^{(3)})$

Иначе

$g_{kj}^{(l)} = (v_k, v_j)$ ,  $g = g_{kj}^{(l)}$

Шаг 2.5.2.

Если  $j < k$

$\gamma_{jk} = \left( g - \sum_{i=1}^{j-1} \gamma_{ik} \gamma_{ij} \right) / \gamma_{jj}$

Иначе

$\gamma_{kk} = \sqrt{g - \sum_{i=1}^{k-1} \gamma_{ik}^2}$

Шаг 2.6.  $c_k = \left( (b^{(m)}, v_k) - \sum_{i=1}^{k-1} \gamma_{ik} c_i \right) / \gamma_{kk}$

Шаг 2.7.  $\rho^2 = \rho^2 - c_k^2$

Шаг 2.8. Если  $k > n$

$r = r - \frac{c_k}{\gamma_{kk}} \sum_{i=1}^{k-1} \gamma_{ik} v_i - c_k v_k$

Шаг 2.9. Если  $k = n$

$d = \Gamma^{-1}c$ ,  $r = r - \sum_{i=1}^k d_i v_i$

Шаг 3.  $c = \Gamma^{-1}c$ ,  $x^{(m)} = \sum_{i=1}^k c_i u_i$ .

Здесь переменные  $s_k^{(l)}$  задают значения параметра  $s$ , для которых определяются элементы матриц  $g_{kj}^{(1)}, g_{kj}^{(2)}, g_{kj}^{(3)}$ . Так как матрица Грама симметричная, то в памяти программы достаточно хранить только верхнюю треугольную часть матрицы.

#### 4. Численный эксперимент

Работоспособность и эффективность предложенных алгоритмов иллюстрируется на примере их использования в периодическом малосигнальном анализе. Типичным применением этого анализа является моделирование схемы смесителя, на входы которой подаются два сигнала разной частоты  $\Omega$ ,  $\omega$ . Один из этих сигналов считается большим, а второй — малым. В отличие от традиционного малосигнального анализа в периодическом малосигнальном анализе вначале определяется периодический установившийся режим по большому сигналу, а затем рассчитывается реакция схемы на малое синусоидальное воздействие, частота которого должна варьироваться в заданном диапазоне. В периодическом малосигнальном анализе на основе гармонического баланса [15] периодический установившийся режим задается значениями гармоник частоты большого сигнала с индексами  $k = -K_h, \dots, K_h$ , а линейная система решается относительно комбинационных частот с индексами  $[k, +1]$ , где 1 соответствует первой гармонике малого сигнала. Линейная система имеет вид [15]

$$A(\omega)x(\omega) = b(\omega), \quad (35)$$

где  $\omega$  — частота малого сигнала. Размерность линейной системы  $(2K_h + 1)N$ , где  $N$  — число узловых переменных схемы.

Периодический малосигнальный анализ позволяет рассчитать зависимости комбинационных компонент отклика схемы  $|\omega + k\Omega|$  от частоты входного малосигнального воздействия  $\omega$ . Ниже приведены результаты анализа для трех схем, характеристики которых представлены в табл. 1.

Схема 1 — это одностранзисторный смеситель, который содержит восемь элементов (один транзистор, два резистора, два конденсатора, одна индуктивность и два источника напряжения). На схему подаются два входных сигнала с частотой 1 МГц, амплитудой 0,03 В и с частотой 95...105 кГц, амплитудой 0,001 В. Преобразователь частоты (схема 2) содержит

Таблица 1

Схема	Число элементов	$N$	$K_h$	$(2K_h + 1)N$
1	8	11	30	671
2	19	16	30 60	976 1936
3	99	59	10 30 60	1239 3599 7139

19 элементов. Входные сигналы имеют частоту 140 МГц с амплитудой 0,5 В и амплитудой 0,0125 В с частотой из диапазона 9,5...10,5 МГц. Схема 3 — смеситель на основе ячейки Гильберта, последовательно с которым включены фильтр и широкополосный усилитель. Эта схема содержит 17 транзисторов, 47 резисторов, 30 конденсаторов и 5 индуктивностей. Входной сигнал имеет частоту 1 ГГц и амплитуду 0,4 В, частота малосигнального воздействия изменяется в диапазоне 860...900 МГц.

Вычислительные затраты периодического малосигнального анализа определяются затратами на решение систем линейных уравнений (35). В итерационных методах рассматриваемого класса эти затраты в основном зависят от затрат на матрично-векторное умножение и операцию ортогонализации векторов. В табл. 2 приведено количество матрично-векторных умножений для метода GMRES, ММН, ММН-Г (3—4 столбцы). Как следует из этой таблицы (столбцы 6, 7), предложенные алгоритмы решения позволяют снизить общее число матрично-векторных умножений в 30—40 (ММН) или в 12—35 (ММН-Г) раз.

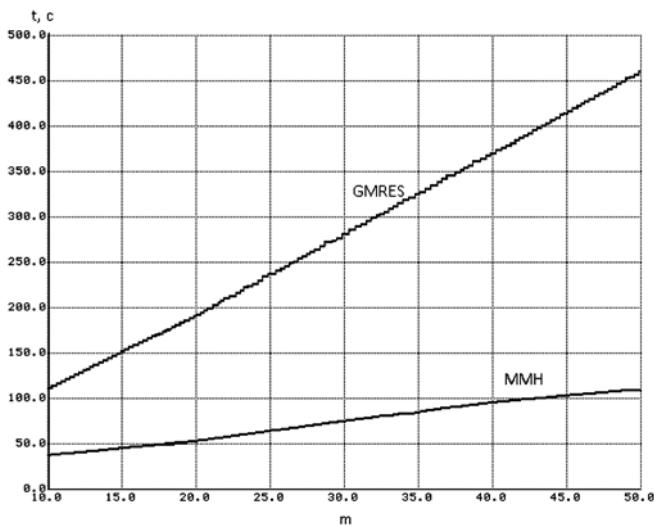
Вместе с тем суммарные затраты на построение ортонормированного базиса в алгоритме ММН выше, чем в алгоритме GMRES, так как число векторов в алгоритме ММН больше, чем их число в алгоритме GMRES в пересчете на одну частотную точку (столбцы 3, 4 в табл. 3). Вследствие этого суммарный вычислительный выигрыш определяется соотношением трудоемкости операций матрично-векторного умножения и ортогонализации. Для схемы 3 суммарный выигрыш показан в табл. 3 (столбец 6).

Таблица 2

Схема	$K_h$	#mv_gmres	#mv_ммн	#mv_ммн-г	#mv_gmres / #mv_ммн	#mv_gmres / #mv_ммн-г
1	30	510	18	41	28	12
2	30 60	612 625	19 21	31 32	32 29,8	19,7 19,5
3	10 60	1836 1836	42 45	48 52	43,7 40,8	38 35

Таблица 3

Схема	$K_n$	$t_{ort\_gmres},$ с	$t_{ort\_ммн},$ с	$t_{ort\_ммн}/$ $t_{ort\_gmres}$	$t_{gmres}/$ $t_{ммн}$
3	10	5,1	16	3,1	4,1
	60	15	41	2,7	3,5



Зависимость вычислительных затрат в алгоритме GMRES и MMH от числа значений частоты малосигнального возбуждения

В алгоритме MMH-Г затраты на построение ортонормированного базиса отсутствуют, поэтому вычислительный выигрыш здесь может быть выше. В частности, для схемы 3 выигрыш составляет 5—6 раз. Однако, как показали эксперименты, при увеличении амплитуды большого сигнала может наблюдаться расходимость метода.

Число операций матрично-векторного умножения, вычислительные затраты на их выполнение и ортогонализацию векторов зависят от числа точек изменения частоты малосигнального возбуждения, а также от ширины диапазона ее изменения. На рисунке представлены вычислительные затраты в зависимости от числа значений частоты малого сигнала.

### Заключение

В работе рассмотрены итерационные методы решения параметризованной системы линейных алгебраических уравнений с многими правыми частями. Предложены алгоритмы, построенные на использовании результатов матрично-векторного умножения и повторного использования найденных векторов. Показано, что эти алгоритмы применимы для решения линейных систем с матрицами, коэффициенты которых линейно зависят от скалярного параметра.

Предложенные алгоритмы используются для получения эффективной вычислительной процедуры периодического малосигнального анализа на основе гармонического баланса. Проведенные численные эксперименты по моделированию нелинейных электронных схем показали, что вычислительные затраты определяются операциями матрично-векторного умножения и построения ортонормированного базиса векторов. Предложенный алгоритм MMH с ортогонализацией позволяет значительно снижать число операций умножения при некотором росте затрат на ортогонализацию. Алгоритм MMH без ортогонализации позволяет получить больший вычислительный выигрыш, но его применимость ограничена вследствие возможной расходимости алгоритма.

### Список литературы

1. Saad Y. Iterative methods for Sparse Linear Systems. PWS publishing company, 1996.
2. Freund R. W. A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems // SIAM J. Sci. Stat. Comput. 1993. Vol. 14, N. 2. P. 470—482.
3. Saad Y., Schultz M. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems // SIAM J. Sci. and Statist. Comput. 1986. Vol. 7, N. 3. P. 856—869.
4. Gourary M. M., Rusakov S. G., Ulyanov S. L., Zharov M. M. et al. Iterative Solution of Linear Systems in Harmonic Balance Analysis // IEEE MTT-S Int. Microwave Symposium Digest. 1997. P. 1507—1510.
5. Boyse W. E., Seidel A. A. A block QMR method for computing multiple simultaneous solutions to complex symmetric systems // SIAM J. Sci. Comput. 1996. Vol. 17, N. 1. P. 263—274.
6. Freund R. and Malhotra M. A block QMR algorithm for non-Hermitian linear systems with multiple right-hand sides // Linear Alg. Appl. 1997. Vol. 254, N. 1—3. P. 197—257.
7. Simoncini V., Gallopoulos E. An iterative method for nonsymmetric systems with multiple right-hand sides // SIAM J. Sci. Comput., 1995. Vol. 16, N. 4. P. 917—933.
8. Simoncini V., Gallopoulos E. A hybrid block GMRES method for nonsymmetric systems with multiple right-hand sides // J. Comput. Appl. Math. 1996. Vol. 66, N. 1. P. 457—469.
9. Baker A. H., Dennis J. M., Jessup E. R. On improving linear solver performance: A block variant of GMRES // SIAM J. Sci. Comput. 2006. Vol. 27, N. 5. P. 1608—1626.
10. Simoncini V. Restarted full orthogonalization method for shifted linear systems // BIT Numer. Math. 2003. Vol. 43, N. 2. P. 459—466.
11. Gu G-D., Simoncini V. Numerical solution of parameter-dependent linear systems // Linear Alg. Appl. 2005. Vol. 12, N. 9. P. 923—940.
12. Baumann M., Gijzen M. B. Nested Krylov methods for shifted linear systems // SIAM J Sci Comput. 2015. Vol. 37, N. 5. P. S90—S112.
13. Darnell D., Morgan R. B., Wilcox W. Deflated GMRES for systems with multiple shifts and multiple right-hand sides // Linear Algebra Appl. 2008. Vol. 429, N. 10. P. 2415—2434.
14. Вержбицкий В. М. Численные методы (математический анализ и обыкновенные дифференциальные уравнения). М.: Высшая школа, 2001.
15. Гурарий М. М., Жаров М. М., Русаков С. Г., Ульянов С. Л. Метод малосигнального анализа для моделирования многочастотных радиотехнических схем // В сб. трудов Всероссийской научно-технической конференции "Проблема разработки перспективных микроэлектронных систем — 2008" (МЭС-2008). М.: ИППМ РАН, 2008. С. 71—76.

## The Iterative Method for Solving Parameterized System of Linear Equations with Multiple Right Hand Sides

*The problem of solving a system of linear equations with multiple right hand sides and matrix dependent on the parameter is considered. The iterative method with reuse of the previous results of matrix vector multiplications is suggested. The method provides the reduction of computational efforts and decrease of matrix-vector multiplications due to reuse of computed vectors. The proposed method is a generalization of known Krylov subspace algorithms. The application of proposed algorithms for periodic small-signal analysis of nonlinear electronic circuits is shown.*

**Keywords:** system of linear algebraic equations, linear transform, orthogonalization, iterative methods, Krylov subspace, approximate solution

### References

1. Saad Y. *Iterative methods for Sparse Linear Systems*. Boston, PWS publishing company, 1996.
2. Freund R. W. A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems, *SIAM J. Sci. Stat. Comput.* 1993, vol. 14, no. 2, pp. 470–482.
3. Saad Y., Schultz M. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. and Statist. Comput.*, 1986, vol. 7, no. 3, pp. 856–869.
4. Gourary M. M., Rusakov S. G., Ulyanov S. L., Zharov M. M. et al. Iterative Solution of Linear Systems in Harmonic Balance Analysis, *IEEE MTT-S Int. Microwave Symposium Digest*, 1997, pp. 1507–1510.
5. Boyse W. E., Seidel A. A. A block QMR method for computing multiple simultaneous solutions to complex symmetric systems, *SIAM J. Sci. Comput.*, 1996, vol. 17, no. 1, pp. 263–274.
6. Freund R., Malhotra M. A block QMR algorithm for non-Hermitian linear systems with multiple right-hand sides, *Linear Alg. Appl.*, 1997, vol. 254, no. 1–3, pp.197–257.
7. Simoncini V., Gallopoulos E. An iterative method for non-symmetric systems with multiple right-hand sides, *SIAM J. Sci. Comput.*, 1995, vol. 16, no. 4, pp. 917–933.
8. Simoncini V., Gallopoulos E. A hybrid block GMRES method for nonsymmetric systems with multiple right-hand sides, *J. Comput. Appl. Math.*, 1996, vol. 66, no. 1, pp. 457–469.
9. Baker A. H., Dennis J. M., Jessup E. R. On improving linear solver performance: A block variant of GMRES, *SIAM J. Sci. Comput.*, 2006, vol. 27, no. 5, pp. 1608–1626.
10. Simoncini V. Restarted full orthogonalization method for shifted linear systems, *BIT Numer. Math.*, 2003, vol. 43, no. 2, pp. 459–466.
11. Gu G-D., Simoncini V. Numerical solution of parameter-dependent linear systems, *Linear Alg. Appl.*, 2005, vol. 12, no. 9, pp. 923–940.
12. Baumann M., Gijzen M. B. Nested Krylov methods for shifted linear systems, *SIAM J Sci Comput.*, 2015, vol. 37, no. 5, pp. S90–S112.
13. Darnell D., Morgan R. B., Wilcox W. Deflated GMRES for systems with multiple shifts and multiple right-hand sides, *Linear Algebra Appl.*, 2008, vol. 429, no. 10, pp. 2415–2434.
14. Verbitsky V. M. *Chislennyye metody (mathematischeskiy analiz i jbknovennyye differentsialnyye uravneniya)*, Moscow: Vysshaya shkola, 2001 (in Russian).
15. Gourary M. M., Zharov M. M., Rusakov S. G., Ulyanov S. L. Metod malosignalnogo analiza dla modelirovaniya mnogochastotnykh radiotekhnicheskikh shem, *Sb. trudov Vserossiyskoy nauchno-tekhnicheskoy konferentsii "Problemy razrabotki perspektivnykh mikroelektronnykh shem"*. Moscow: IPPM RAN, 2008, pp. 71–76 (in Russian).

УДК 519.6

DOI: 10.17587/it.24.370-386

T. A. Агасиев, аспирант, e-mail: agtaleh@mail.ru,  
А. П. Карпенко, д-р физ.-мат. наук, проф., зав. кафедрой, e-mail: arkarpenko@mail.ru,  
Московский государственный технический университет им. Н. Э. Баумана

## Современные техники глобальной оптимизации. Обзор

*Представляем постановку задачи глобальной условной оптимизации, основной особенностью которой является высокая вычислительная сложность целевой функции. Даем определения таких сущностей, как характерные признаки задачи, базовые задача и алгоритм оптимизации, метазадача и метаалгоритм оптимизации, стратегия базовой задачи, индикатор эффективности стратегии. Определяем и приводим постановки мультииндикаторной, мультиклассовой и мультибюджетной задач метаоптимизации. На основе анализа около 100 публикаций даем обзор современных методов ландшафтного анализа целевых функций, а также методов метаоптимизации базовых алгоритмов. Наконец, представляем краткий обзор современного программного обеспечения, используемого для решения базовых и метазадач оптимизации.*

**Ключевые слова:** глобальная оптимизация, ландшафтный анализ, метаоптимизация, суррогатное моделирование, анализ эффективности алгоритмов оптимизации

### Введение

Задачи глобальной оптимизации возникают во многих приложениях, например, при раз-

работке комплексных инженерных изделий. Поэтому современные системы автоматизированного проектирования (САД) и инженерного анализа (САЕ) включают в себя программы,