

УДК 519.615,519.63

БЕЗМАТРИЧНЫЕ ИТЕРАЦИОННЫЕ ПРОЦЕССЫ СО СРЕДНЕКВАДРАТИЧНЫМ ПОДАВЛЕНИЕМ ОШИБКИ ДЛЯ БОЛЬШИХ СИСТЕМ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

И. В. БОНДАРЬ¹⁾, Б. В. ФАЛЕЙЧИК¹⁾

¹⁾Белорусский государственный университет, пр. Независимости, 4, 220030, г. Минск, Беларусь

Рассмотрены итерационные процессы решения больших систем нелинейных уравнений, не требующие хранения и факторизации матрицы Якоби. Для ускорения сходимости в случае большого спектрального числа обусловленности этой матрицы предлагается специальная техника среднеквадратичного подавления ошибки, реализация которой требует решения линейной задачи наименьших квадратов небольшой размерности. В линейном случае полученный метод схож с предобусловленным обобщенным методом минимальных невязок. В нелинейном же случае в отличие от популярного безматричного метода Ньютона – Крылова разработанный подход не содержит операции разностной аппроксимации производной. Проведены вычислительные эксперименты на трех системах нелинейных уравнений, возникающих в результате конечно-разностной аппроксимации двумерных уравнений в частных производных эллиптического типа. Показано преимущество разработанного подхода по сравнению с методом Ньютона – Крылова на рассмотренных тестовых задачах.

Ключевые слова: системы нелинейных уравнений; безматричные методы; ускорение сходимости; метод наименьших квадратов; метод Ньютона – Крылова; разностные схемы.

Образец цитирования:

Бондарь И. В., Фалейчик Б. В. Безматричные итерационные процессы со среднеквадратичным подавлением ошибки для больших систем нелинейных уравнений // Журн. Белорус. гос. ун-та. Математика. Информатика. 2017. № 3. С. 73–84.

For citation:

Bondar I. V., Faleichyk B. V. Matrix-free iterative processes with least-squares error damping for nonlinear systems of equations. *J. Belarus. State Univ. Math. Inform.* 2017. No. 3. P. 73–84 (in Russ.).

Авторы:

Иван Васильевич Бондарь – ассистент кафедры вычислительной математики факультета прикладной математики и информатики.

Борис Викторович Фалейчик – кандидат физико-математических наук; доцент кафедры вычислительной математики факультета прикладной математики и информатики.

Authors:

Ivan V. Bondar, assistant at the department of computational mathematics, faculty of applied mathematics and computer science.

bondarivanv@gmail.com

Barys V. Faleichyk, PhD (physics and mathematics); associate professor at the department of computational mathematics, faculty of applied mathematics and computer science.

faleichik@gmail.com

MATRIX-FREE ITERATIVE PROCESSES WITH LEAST-SQUARES ERROR DAMPING FOR NONLINEAR SYSTEMS OF EQUATIONS

I. V. BONDAR^a, B. V. FALEICHYK^a

^aBelarusian State University, 4 Niezaliežnasci Avenue, Minsk 220030, Belarus
Corresponding author: I. V. Bondar (bondarivanv@gmail.com)

New iterative processes for numerical solution of big nonlinear systems of equations are considered. The processes do not require factorization and storing of Jacobi matrix and employ a special technique of convergence acceleration which is called least-squares error damping and requires solution of auxiliary linear least-squares problems of low dimension. In linear case the resulting method is similar to the general minimal residual method (GMRES) with preconditioning. In nonlinear case, in contrast to popular Newton – Krylov method, the computational scheme do not involve operation of difference approximation of derivative operator. Numerical experiments include three nonlinear problems originating from two-dimensional elliptic partial differential equations and exhibit advantage of the proposed method compared to Newton – Krylov method.

Key words: nonlinear systems of equations; matrix-free methods; acceleration of convergence; least-squares; Newton – Krylov method; difference schemes.

Задача решения больших систем нелинейных уравнений часто возникает при дискретизации дифференциальных уравнений. Решение таких систем ньютоновскими методами затруднено необходимостью строить и обращать матрицу Якоби (или некоторую аппроксимацию к ней). В работе [1] для случая жесткой системы обыкновенных дифференциальных уравнений предложен метод решения систем нелинейных уравнений, не требующий хранения и обращения матриц и обладающий улучшенными свойствами сходимости по сравнению с обычным методом простой итерации. При этом сходимость существенно замедляется в случае, когда матрица Якоби системы имеет сильно различающиеся по модулю собственные значения (содержит большое спектральное число обусловленности $\kappa(A) = \rho(A)\rho(A^{-1})$). В настоящей работе рассматриваются схожие по структуре итерационные процессы, сходимость которых в случае большого $\kappa(A)$ ускоряется за счет специальной процедуры среднеквадратичного подавления ошибки. Итоговый метод в линейном случае схож с итерационными методами, основанными на подпространствах Крылова [2], и легко обобщается на нелинейный случай. При этом в отличие от популярного безматричного метода Ньютона – Крылова (НК) [3] алгоритм не содержит чувствительной к вычислительным погрешностям операции разностной аппроксимации производной и, как показали вычислительные эксперименты, является более эффективным на рассмотренных тестовых задачах.

Базовый итерационный процесс

Описание метода начнем с линейного случая. Рассмотрим систему

$$Ax = b, \quad (1)$$

где A – вещественная матрица $n \times n$ со спектром $\Sigma(A) \subset [-M, -\mu] \subset \mathbb{R}^-$, $0 < \mu < M$, $b \in \mathbb{R}^n$. Стандартным образом приведем систему (1) к виду

$$x = Bx + g, \quad (2)$$

где $B = I + \omega A$, $g = -\omega b$, $\omega > 0$, – некоторый масштабирующий множитель. В дальнейшем будем предполагать $\omega \in (0, 2M^{-1})$, что дает $\Sigma(B) \subset (-1, 1)$. В [4, § 86.8] описаны так называемые универсальные алгоритмы, представляющие собой двухшаговые итерационные процессы для решения систем вида (2):

$$x^{j+1} = \alpha_{j+1}(Bx^j + g) + \beta_{j+1}x^j + \gamma_{j+1}x^{j-1}, \quad j = 0, 1, 2, \dots \quad (3)$$

По построению процесса невязка

$$r(x) = Bx + g - x = \omega(Ax - b) \quad (4)$$

на приближениях x^j представима в виде

$$r(x^j) = e_j(B)r(x^0),$$

где $\{e_j\}$ – система многочленов, удовлетворяющая аналогичному (3) рекуррентному соотношению

$$e_{j+1}(t) = (\alpha_{j+1}t + \beta_{j+1})e_j(t) + \gamma_{j+1}e_{j-1}(t), \quad j = 0, 1, 2, \dots, \quad e_0(t) = 1.$$

Если коэффициенты процесса определены как

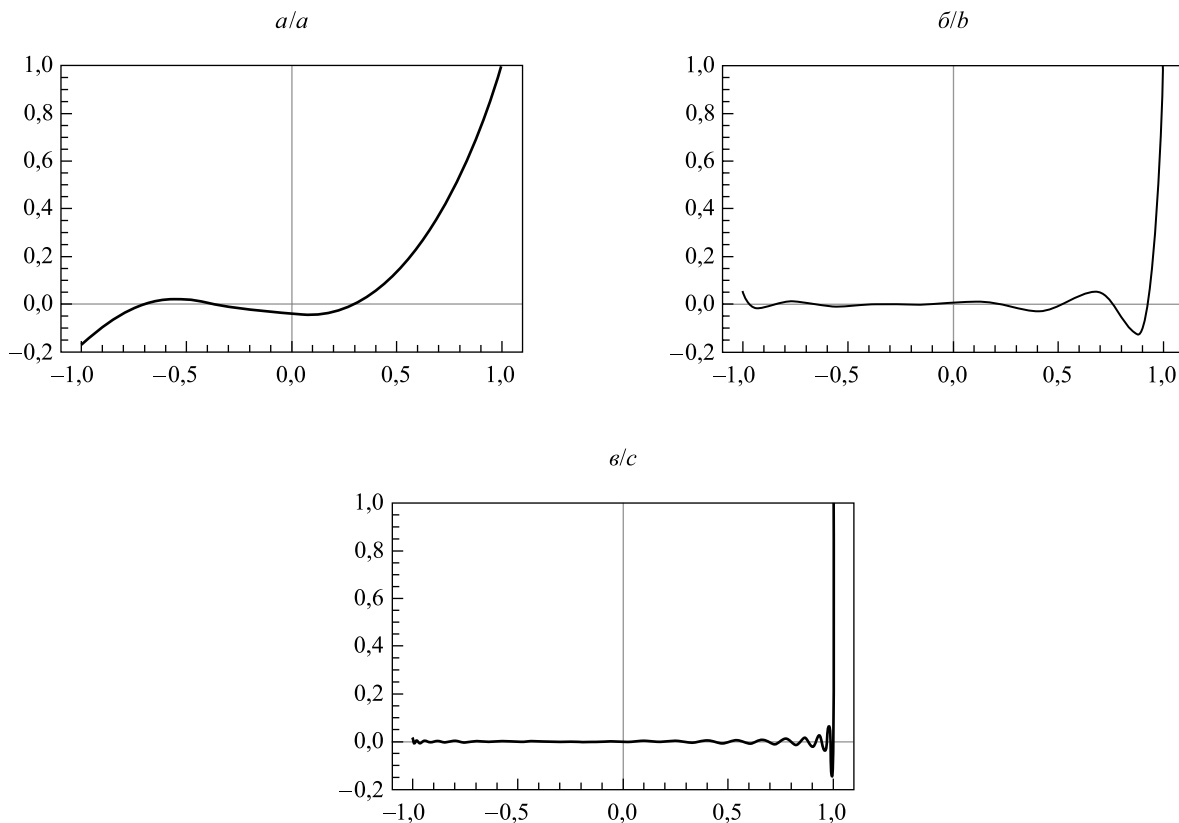
$$\alpha_1 = \frac{3}{4}, \quad \beta_1 = \frac{1}{4}, \quad \gamma_1 = 0, \quad (5)$$

$$\alpha_j = \frac{j(2j+1)}{(j+1)^2}, \quad \beta_j = \frac{j}{(2j-1)(j+1)^2}, \quad \gamma_j = -\frac{(2j+1)(j-1)^2}{(2j-1)(j+1)^2}, \quad j = 2, 3, \dots,$$

то соответствующее множество $\{e_j\}_{j=0}^n$ образует систему ортогональных многочленов на отрезке $[-1, 1]$ по весу $\rho(t) = 1 - t$ [4, § 93]. Основным же их свойством является следующее:

$$e_j = \operatorname{argmin}_{p \in \mathbb{P}_1^j} \int_{-1}^1 |p(t)|^2 dt,$$

где \mathbb{P}_1^j – множество многочленов p степени не выше j таких, что $p(1) = 1$. В связи с этим в [4] двухшаговые процессы с таким выбором коэффициентов называются наилучшими в среднем. Графики e_3, e_{10}, e_{40} приведены на рис. 1.



Графики многочленов: $a - e_3; б - e_{10}; в - e_{40}$
Graphs of polynomials: $a - e_3; б - e_{10}; в - e_{40}$

На практике рассматриваемый процесс (3), как правило, перезапускается через каждые s итераций. Отображение $x^0 \mapsto x^{j+1}$, задаваемое формулой (3), обозначим Φ_{j+1} :

$$\begin{aligned} \Phi_0(x) &= x, \\ \Phi_{j+1}(x) &= \alpha_{j+1}\varphi(\Phi_j(x)) + \beta_{j+1}\Phi_j(x) + \gamma_{j+1}\Phi_{j-1}(x), \quad j = 0, 1, \dots, s-1, \end{aligned} \quad (6)$$

где $\varphi(x) = Bx + g$. В общем случае (см. далее) φ будет некоторой нелинейной вектор-функцией. Используя (6), полученный в результате «зацикливания» после s итераций процесс (3) можно записать в виде

$$x^{k+1} = \Phi_s(x^k), \quad k = 0, 1, \dots \quad (7)$$

Для невязки (4) процесса (7) по построению имеем $r(x^k) = e_s(B)^k r(x^0)$. Согласно известному критерию сходимости итерационных процессов решения систем линейных алгебраических уравнений (СЛАУ) процесс (7) сходится, начиная с произвольного начального приближения, тогда и только тогда, когда

$$\rho(e_s(B)) = \max_{v \in \Sigma(B)} |e_s(v)| < 1.$$

При этом для всех достаточно больших k имеет место оценка погрешности

$$\|x^k - x^*\| \leq \rho(e_s(B))^k \|x^0 - x^*\|. \quad (8)$$

Поскольку $|e_j(t)| < 1$ при всех $t \in [-1, 1], j > 0$, процесс (7), (6) с коэффициентами (5) будет сходящимся при любых s , если $\Sigma(B) \subset [-1, 1]$. Однако при больших значениях $\rho(A)\rho(A^{-1})$ максимальное по модулю собственное значение v_1 матрицы B будет очень близко к единице. Тогда в силу равенства $e_j(1) = 1$ будем иметь $\rho(e_s(B)) = |e_s(v_1)| \approx 1$, т. е. сходимость процесса будет очень медленной.

Нелинейный случай

Рассмотренные выше итерационные процессы естественным образом обобщаются на случай системы нелинейных уравнений вида

$$f(x) = 0, \quad f: \mathbb{R}^n \rightarrow \mathbb{R}^n. \quad (9)$$

Для этого в формулах (7), (6) положим

$$\varphi(x) = x + \omega f(x), \quad (10)$$

где $\omega \in \mathbb{R}$. В результате получим итерационный процесс для решения уравнения $x = \varphi(x)$, которое имеет те же корни, что и (9). В дальнейшем будем предполагать, что функция f дифференцируема по Фреше.

Лемма 1. Пусть функция φ дифференцируема по Фреше в некоторой окрестности своей неподвижной точки x^* . Если коэффициенты рекуррентной формулы (6) удовлетворяют условию

$$\alpha_j + \beta_j + \gamma_j = 1 \quad (11)$$

для $j = 1, \dots, s-1$, то

$$\Phi'_j(x^*) = e_j(J^*), \quad j = 0, \dots, s,$$

где $J^* = \varphi'(x^*)$.

Доказательство. Условие (11) эквивалентно $\Phi_j(x^*) = x^*$, $j = 1, \dots, s$. По определению легко проверить утверждение леммы для $j = 0$ и $j = 1$. Для остальных j по индукции будем иметь $\Phi'_{j+1}(x^*) = \alpha_{j+1}J^*e_j(J^*) + \beta_{j+1}e_j(J^*) + \gamma_{j+1}e_{j-1}(J^*) = e_{j+1}(J^*)$.

Лемма доказана.

Пусть матрица $f'(x^*)$ обладает теми же спектральными свойствами, что и матрица A из СЛАУ (1): $\Sigma(f'(x^*)) \subset [-M, -\mu]$. Тогда при достаточно малых значениях параметра ω для $\varphi(x) = x + \omega f(x)$

будем иметь $\Sigma(J^*) = \Sigma(I + \omega f'(x^*)) \subset (-1, 1)$ и, как следствие, $\rho(\Phi'_s(x^*)) = \rho(e_s(J^*)) < 1$ по лемме 1. В результате по теореме Островского [5, теорема 10.1.3] получим, что при указанных выше ограничениях итерационный процесс (7), (6), (10), с коэффициентами (5) будет порождать сходящуюся к x^* последовательность $\{x^k\}$ при всех x^0 из некоторой окрестности $\Omega(x^*)$. При этом согласно теореме о линейной сходимости [5, теорема 10.1.4] множитель сходимости такой последовательности, определяемый как $R_1\{x^k\} = \limsup_{k \rightarrow \infty} \|x^k - x^*\|^{1/k}$, по аналогии с (8) будет равен

$$R_1\{x^k\} = \rho(\Phi'_s(x^*)) = \rho(e_s(\varphi(x^*))) = \max_{v \in \Sigma(\varphi'(x^*))} |e_s(v)|.$$

Таким образом, как и в линейном случае, при больших значениях спектрального числа обусловленности матрицы $f'(x^*)$ сходимость рассматриваемых итерационных процессов будет медленной.

Среднеквадратичное подавление ошибки

Для ускорения сходимости процессов (7), (6) обратимся снова к линейному случаю: $\varphi(x) = Bx + g$. Пусть B в (2) – диагонализируемая матрица с собственными значениями $\{v_i\}$, $-1 \leq v_n \leq v_{n-1} \leq \dots \leq v_2 \leq v_1 < 1$, и линейно независимыми собственными векторами $\{\xi_i\}$. Невязку на k -й итерации обозначим

$$r^k = r(x^k) = (B - I)x^k + g.$$

Разложим r^0 по базису $\{\xi_i\}$: $r^0 = \sum_{i=1}^n a_i \xi_i$, $a_i \in \mathbb{R}$. Тогда по построению $r^k = \sum_{i=1}^n a_i (e_s(v_i))^k \xi_i$. В силу свойств многочлена e_s после достаточно большого числа итераций K (или при достаточно большом значении s) в разложении невязки r^K будут доминировать слагаемые, соответствующие близким к 1 собственным значениям:

$$r^K \approx \sum_{i=1}^m d_i \xi_i, \quad m < n, \quad d_i = a_i (e_s(v_i))^K. \quad (12)$$

Все последующие итерации будут лишь незначительно уменьшать невязку, так как i -я компонента этого разложения при $i \leq m$ будет умножаться на $e_s(v_i) \approx 1$. Для того чтобы исправить ситуацию, положим $x^0 = x^K$, выполним m итераций (7) и построим уточненное приближение по формуле

$$\hat{x} = \sum_{k=0}^m c_k x^k,$$

исходя из условий $\|r(\hat{x})\|_2 \rightarrow \min$ и $\sum_{k=0}^m c_k = 1$. Последнее условие при $x^0 = x^*$ гарантирует выполнение тождества $\hat{x} = x^*$, а также дает $r(\hat{x}) = \hat{r} = \sum_{k=0}^m c_k r^k$. Тогда задача нахождения коэффициентов $\{c_k\}$ принимает вид

$$\left\| \sum_{k=0}^m c_k r^k \right\|_2 \rightarrow \min, \quad \sum_{k=0}^m c_k = 1. \quad (13)$$

Решив эту задачу, полагаем $x^0 = \hat{x}$ и в случае необходимости повторяем описанную процедуру. Этот прием в дальнейшем будем называть *среднеквадратичным подавлением ошибки*.

Обсудим некоторые свойства уточненного приближения \hat{x} . Во-первых, по построению имеем $\|\hat{r}\|_2 \leq \|r^k\|_2$, $0 \leq k \leq m$. Во-вторых, покажем, что если в (12) имеет место точное равенство, то процедура среднеквадратичного подавления даст точное решение системы.

Лемма 2. Пусть $r^0 = \sum_{i=1}^m a_i \xi_i$. Тогда существует такой набор коэффициентов $\{c_k\}_{k=0}^m$, что $\sum_{k=0}^m c_k = 1$ и $\hat{r} = \sum_{k=0}^m c_k r^k = 0$.

Доказательство. По построению итерационного процесса имеем $r^k = e_s(B)^k r^0$. Тогда

$$\hat{r} = \sum_{k=0}^m c_k r^k = \sum_{k=0}^m c_k e_s(B)^k r^0 = C_m(e_s(B))r^0 = \sum_{i=1}^m a_i C_m(\lambda_i) \xi_i,$$

где $\lambda_i = e_s(v_i)$; $C_m(t) = c_0 + c_1 t + \dots + c_m t^m$. Если выбрать коэффициенты $\{c_k\}$ так, что $C_m(t) = K(t - \lambda_1) \dots (t - \lambda_m)$, то будем иметь $\hat{r} = \sum_{i=1}^m a_i C_m(\lambda_i) \xi_i = 0$. Условие нормировки $\sum_{k=0}^m c_k = 1$ можно выполнить, положив $K = \prod_{i=1}^m (1 - \lambda_i)^{-1}$.

Лемма доказана.

В общем случае, конечно, $\hat{r} \neq 0$, однако, если в разложении r^0 по собственным векторам $\{\xi_i\}$ доминируют m слагаемых, мы вправе ожидать, что $\hat{x} \approx x^*$.

Для решения задачи (13) на практике выразим $c_m = 1 - \sum_{k=0}^{m-1} c_k$ и получим обычную линейную задачу наименьших квадратов

$$\left\| \sum_{k=0}^{m-1} c_k (r^k - r^m) + r^m \right\|_2 \rightarrow \min. \quad (14)$$

Отметим, что схожая процедура ускорения сходимости в случае линеаризованной задачи использована в работе [6]. Далее, будем применять ее напрямую для нелинейных задач.

Исследуем корректность задачи (14) в линейном случае. Рассмотрим матрицу

$$\Delta_m^L = \begin{bmatrix} 1 - \lambda_1^m & \lambda_1 - \lambda_1^m & \dots & \lambda_1^{m-1} - \lambda_1^m \\ 1 - \lambda_2^m & \lambda_2 - \lambda_2^m & \dots & \lambda_2^{m-1} - \lambda_2^m \\ \vdots & \vdots & \dots & \vdots \\ 1 - \lambda_L^m & \lambda_L - \lambda_L^m & \dots & \lambda_L^{m-1} - \lambda_L^m \end{bmatrix}, \quad \lambda_i \in \mathbb{R}. \quad (15)$$

Лемма 3.

$$\det \Delta_m^L = (-1)^m \prod_{i=1}^m (\lambda_i - 1) \prod_{1 \leq j < i \leq m} (\lambda_i - \lambda_j). \quad (16)$$

Доказательство. Если $\lambda_i = 1$ хотя бы для одного i , то матрица Δ_m^L вырождена и $\det \Delta_m^L = 0$. В дальнейшем предполагаем, что $\lambda_i \neq 1$, $i = 1, \dots, m$. Вычтем из i -го столбца матрицы Δ_m^L предыдущий столбец для $i = m, \dots, 2$. Тогда

$$\begin{aligned} \det \Delta_m^L &= \det \begin{bmatrix} 1 - \lambda_1^m & \lambda_1 - 1 & \dots & \lambda_1^{m-2} - \lambda_1^{m-3} & \lambda_1^{m-1} - \lambda_1^{m-2} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 1 - \lambda_m^m & \lambda_m - 1 & \dots & \lambda_m^{m-2} - \lambda_m^{m-3} & \lambda_m^{m-1} - \lambda_m^{m-2} \end{bmatrix} = \\ &= \prod_{i=1}^m (\lambda_i - 1) \det \tilde{W}, \quad \tilde{W} = \begin{bmatrix} (1 - \lambda_1^m)/(\lambda_1 - 1) & 1 & \dots & \lambda_1^{m-3} & \lambda_1^{m-2} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ (1 - \lambda_m^m)/(\lambda_m - 1) & 1 & \dots & \lambda_m^{m-3} & \lambda_m^{m-2} \end{bmatrix}. \end{aligned}$$

Учитывая, что $\frac{1-\lambda_i^m}{\lambda_i-1} = -\sum_{k=0}^{m-1} \lambda_i^k$, и последовательно добавляя к первому столбцу матрицы \tilde{W} все остальные столбцы, получим следующую матрицу:

$$W = \begin{bmatrix} -\lambda_1^{m-1} & 1 & \dots & \lambda_1^{m-3} & \lambda_1^{m-2} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ -\lambda_m^{m-1} & 1 & \dots & \lambda_m^{m-3} & \lambda_m^{m-2} \end{bmatrix}.$$

Нетрудно заметить, что матрица W при помощи элементарных преобразований может быть приведена к матрице Вандермонда и $\det W = (-1)^m \prod_{1 \leq j < i \leq m} (\lambda_i - \lambda_j)$ [7, с. 40], откуда и получаем (16).

Лемма доказана.

В дальнейшем запись $[u_1 | \dots | u_m]$ будет обозначать матрицу, составленную из вектор-столбцов u_i .

Теорема. Пусть $r^0 = \sum_{i=1}^L a_i \xi_i$, $L \leq n$, $a_i \neq 0$. Тогда, если существуют $m \leq L$ различных значений $e_s(v_i) = \lambda_i \neq 1$, то ранг матрицы $V_m = [r^0 - r^m | \dots | r^{m-1} - r^m]$ равен m , т. е. задача (14) имеет единственное решение.

Доказательство. Обозначим $Z = [a_1 \xi_1 | \dots | a_L \xi_L]$. Тогда $r^k = \sum_{i=1}^L a_i \lambda_i^k \xi_i = Z(\lambda_1^k, \dots, \lambda_L^k)^T$ и $V_m = Z \Delta_m^L$, где Δ_m^L – матрица вида (15). Покажем, что $\text{rank } \Delta_m^L = m$.

Если $L = m$, то утверждение следует из леммы 3. Пусть $m < L$. Не нарушая общности, предположим, что различны первые m чисел λ_i , и рассмотрим матрицу, образованную первыми m строками и столбцами матрицы Δ_m^L . Нетрудно заметить, что эта матрица совпадает с матрицей Δ_m^m (15) и для нее верна лемма 3, откуда по условию имеем $\det \Delta_m^m \neq 0$. Поскольку $\text{rank } Z = L > m$, окончательно получаем $\text{rank } V_m = \text{rank } \Delta_m^L = m$.

Теорема доказана.

В случае нелинейного уравнения (9) теорема, вообще говоря, не будет справедлива, однако задача (14) и техника ее решения ничем не отличаются от линейного случая.

Вычислительные алгоритмы

Процедуру среднеквадратичного подавления ошибки удобно оформить в виде подпрограммы *lsdamp*, которая принимает на вход $m + 1$ вектор x^k и возвращает уточненное приближение \hat{x} (алгоритм 1).

```
lsdamp( $x^0, \dots, x^m$ ):
· for  $k = 0$  to  $m$ :
·   ·  $r^k \leftarrow r(x^k)$ 
·   ·  $V = [r^0 - r^m | \dots | r^{m-1} - r^m]$ 
·   ·  $c = \text{argmin} \|Vc + r^m\|_2$ 
·   · return  $\sum_{j=0}^{m-1} c_j x^j + \left(1 - \sum_{j=0}^{m-1} c_j\right) x^m$ 
```

Алгоритм 1. Среднеквадратичное подавление ошибки
Algorithm 1. Least squares damping of the error

Задачу наименьших квадратов можно решать разными способами [8]. В наших экспериментах использована соответствующая функция библиотеки *NumPy* языка *Python*.

Алгоритм 2 описывает простейший итерационный процесс с использованием среднеквадратичного подавления. На вход здесь передаются начальное приближение x^0 и требуемая точность ε . Отображение Φ_s определяется формулами (6), параметр N_{damp} задает количество последовательных приближений, используемое при расчете уточненного приближения.

```

while  $\|r(x^0)\| > \varepsilon$ 
· for  $k = 1$  to  $N_{\text{damp}}$ 
· ·  $x^k \leftarrow \Phi_s(x^{k-1})$ 
·  $x^0 \leftarrow \text{lsdamp}(x^0, \dots, x^{N_{\text{damp}}})$ 
    
```

Алгоритм 2. Простейший процесс с подавлением ошибки (TSLС + D)

Algorithm 2. Basic two-step process with least squares damping of the error (TSLС + D)

Заметим, что функцию `lsdamp` можно применять для произвольного количества векторов, начиная с двух. Это дает возможность для модификации алгоритма 1: можно проводить подавление ошибки после вычисления каждого нового приближения. Хранение всех последовательных приближений затруднительно, поэтому ограничим их количество параметром N_{damp} . Если в процессе решения придется сделать больше, чем N_{damp} итераций, то векторы приближений, полученные на ранних итерациях, будут замещаться более поздними приближениями. Полученная в результате вычислительная схема приведена в алгоритме 3.

```

 $q = 0$ 
while  $\|r(x^0)\| > \varepsilon$ 
· for  $i = 1$  to  $N_0$ 
· ·  $x^0 \leftarrow \Phi_s(x^0)$ 
·  $x \leftarrow x^0$ 
· for  $k = 0$  to  $N_1$ 
· · if  $q < N_{\text{damp}}$ 
· · ·  $q++$ 
· ·  $x^q \leftarrow \Phi_s(x)$ 
· ·  $x \leftarrow \text{lsdamp}(x^0, \dots, x^q)$ 
· · if  $q == N_{\text{damp}}$ 
· · · for  $p = 1$  to  $N_{\text{damp}}$ 
· · · ·  $x^{p-1} \leftarrow x^p$ 
·  $x^0 \leftarrow x$ 
    
```

Алгоритм 3. Итерационный процесс с модифицированным подавлением ошибки (TSLС + WD)

Algorithm 3. Modified two-step process with «window» damping of the error (TSLС + WD)

Здесь параметры N_0 и N_1 определяют количество обычных итераций базового алгоритма и количество итераций с последующим подавлением соответственно, а N_{damp} – максимальное количество векторов, используемое для подавления.

Метод Ньютона – Крылова

Классический итерационный процесс метода Ньютона для решения системы нелинейных уравнений вида (9) можно записать следующим образом:

$$x^{k+1} = x^k + \delta^k, \quad (17)$$

$$f'(x^k)\delta^k = -f(x^k), \quad k = 0, 1, \dots, \quad (18)$$

где $f'(x^k)$ – матрица Якоби решаемой нелинейной системы. Каждая итерация процесса (17), (18) требует решения СЛАУ той же размерности, что и у нелинейной системы. Применение точных методов

для решения этой задачи сопряжено с известными затруднениями: трудоемкость растет в общем случае пропорционально кубу размерности, вычисление и хранение матрицы Якоби для большой системы может представлять проблему, СЛАНУ (18) может быть плохо обусловленной.

В работе [3] описан подход, позволяющий избавиться от подобных недостатков и получивший название метода Ньютона – Крылова. Суть метода в том, чтобы для решения системы (18) использовать итерационные методы, основанные на подпространствах Крылова. В этом случае основная вычислительная операция – это умножение матрицы Якоби на некоторый вектор v :

$$w = f'(x^k)v. \quad (19)$$

Можно показать, что

$$f'(x)v \approx \frac{f(x + \varepsilon v) - f(x)}{\varepsilon}, \quad (20)$$

где ε – некоторое малое возмущение. Таким образом, для выполнения операции (19) не нужно вычислять и хранить матрицу $f'(x)$, можно воспользоваться (20), что в результате порождает достаточно эффективную вычислительную схему.

Вычислительный эксперимент

Рассмотрим результаты некоторых вычислительных экспериментов. Тестовые системы нелинейных уравнений получены путем дискретизации уравнений в частных производных методом конечных разностей на равномерной сетке по пятиточечному шаблону. Критерий остановки итераций $\|r(x^k)\|_{\infty} \leq 10^{-9}$.

Приведенные ниже задачи 1 и 2 являются задачами с разреженной матрицей Якоби, задача 3 имеет полную матрицу Якоби. В целях оценки спектра матрицы Якоби задач 1 и 3 можно воспользоваться известными выражениями для дискретного лапласиана [9, гл. IV, § 4]. В частности, спектральный радиус этой матрицы, необходимый для нахождения параметра ω , может быть аппроксимирован числом $8h^{-2}$, где h – шаг сетки. Для задачи 2 такие оценки неочевидны, поэтому значение параметра ω подбиралось экспериментально.

Проведено сравнение базового метода (7), (6) (в результатах он обозначается как TSLS – two-step least squares) и описанных алгоритмов с подавлением ошибки (простое подавление ошибки по алгоритму 1 обозначено TSLS + D, по алгоритму 3 – TSLS + WD). В сравнении участвует также метод Ньютона – Крылова.

Для реализации алгоритмов был использован язык *Python* (версия интерпретатора 3.4.3), библиотек *Scipy* (версия 0.13.3) и *Numpy* (версия 1.8.2). Метод Ньютона – Крылова реализован с помощью библиотечной функции *scipy.optimize.newton_krylov*, в качестве «решателя» линейных систем, возникающих в этом методе, был использован алгоритм *lgmres* (*scipy.sparse.linalg.lgmres*) [10]. Задачи наименьших квадратов решались методом на основе сингулярного разложения, реализованным в функции *numpy.linalg.lstsq*.

Для всех алгоритмов на основе процесса (7), (6) было взято значение $s = 100$. Для алгоритмов с подавлением ошибки использованы параметры $N_{\text{damp}} = 14$, $N_0 = 2$, $N_1 = 12$.

В задачах 1 и 2 в качестве точного решения используются функция

$$u_{\text{ex}}(x, y) = \cos(\pi x)\sin(\pi y) + 2 \quad (21)$$

и соответствующие граничные условия.

Задача 1. Рассматривается задача Дирихле для уравнения Пуассона в квадрате $D = [0, 1] \times [0, 1]$:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y, u), \quad (x, y) \in (0, 1) \times (0, 1), \quad (22)$$

$$u(x, y) = 2, \quad (x, y) \in \partial D.$$

В качестве правой части уравнения (22), соответствующей точному решению (21), взята функция

$$f(x, y, u) = -2\pi^2 \cos(\pi x)\sin(\pi y) + e^{-u^2-10} - e^{-u_{\text{ex}}^2-10}.$$

Расчеты проводились с $\omega = 2^{-3}N^{-2}$.

Задача 2. Рассматривается задача Дирихле для нелинейного стационарного дифференциального уравнения с частными производными:

$$\frac{\partial}{\partial x} \left(u^\alpha \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(u^\alpha \frac{\partial u}{\partial y} \right) = f(x, y), \quad (x, y) \in (0, 1) \times (0, 1),$$

$$u(x, y) = 2, \quad (x, y) \in \partial D, \quad (23)$$

где $\alpha > 1$ – некоторый параметр.

Подставив в уравнение (23) точное решение (21), получим вид правой части:

$$f(x, y) = \frac{1}{2} \pi^2 (\cos(\pi x) \sin(\pi y) + 2)^{\alpha-1} \times$$

$$\times (\alpha + \cos(2\pi x) ((\alpha + 1) \cos(2\pi y) - 1) - 8 \cos(\pi x) \sin(\pi y) + \cos(2\pi y) - 1).$$

Для дискретизации воспользуемся интегро-интерполяционным методом [9]. На области D введем сетку $\bar{\omega}_h = \left\{ (x_i, y_j), x_i = ih, y_j = jh, h = \frac{1}{N}, i = 0, N \right\}$ и после стандартных выкладок в итоге получим следующую разностную схему:

$$(u_{i+1,j} - u_{i,j})a_{i+1,j} - (u_{i,j} - u_{i-1,j})a_{i,j} + (u_{i,j+1} - u_{i,j})b_{i,j+1} - (u_{i,j} - u_{i,j-1})b_{i,j} = h^2 f_{i,j},$$

где $a_{i,j} = 2(u_{i,j}^{-\alpha} + u_{i-1,j}^{-\alpha})^{-1}$; $b_{i,j} = 2(u_{i,j}^{-\alpha} + u_{i,j-1}^{-\alpha})^{-1}$. Расчеты проводились с $\alpha = 2,00$, $\omega = 0,04$.

Задача 3. Рассматривается интегро-дифференциальное уравнение

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 10 \left(\int_0^1 \int_0^1 \cosh(u) dx dy \right)^2, \quad (x, y) \in (0, 1) \times (0, 1)$$

с граничными условиями

$$u(x, 0) = 1 - x, \quad u(0, y) = 1 - y, \quad u(1, y) = 0, \quad u(x, 1) = 0.$$

Для дискретизации воспользуемся конечными разностями: на области введем сетку $\bar{\omega}_h = \left\{ (x_i, y_j), x_i = ih_x, y_j = jh_y, h_x = h_y = \frac{1}{N}, i = 0, N \right\}$. Применив простейшую кубатурную формулу для приближения интеграла на сетке, получим следующую систему нелинейных уравнений:

$$N^2 (u_{i-1,j} + u_{i+1,j} - 4u_{i,j} + u_{i,j-1} + u_{i,j+1}) - 10 \left(\frac{1}{N^2} \sum_{p=1}^N \sum_{q=1}^N \cosh(u_{p,q}) \right)^2 = 0,$$

где $i = 1, N - 1, j = 1, N - 1$. Матрица Якоби такой системы, в отличие от предыдущих двух задач, не является разреженной. Расчеты проводились с $\omega = 2^{-3} N^{-2}$.

Результаты вычислительных экспериментов

В таблице результаты представлены в формате *время работы в секундах – количество вычислений правой части уравнения*. При вычислениях по некоторому методу требовалось слишком много времени для сходимости, в ячейке представлен прочерк. На каждой из задач благодаря использованию алгоритмов TSLS + D и TSLS + WD было получено значительное ускорение итерационного процесса по сравнению с базовым методом TSLS, причем с увеличением размерности задачи (n) положительные эффекты от применения среднеквадратичного подавления ошибки усиливались. Для сравнения приведены результаты работы метода Ньютона – Крылова на идентичных задачах. Видно, что алгоритм TSLS + WD на больших размерностях превосходит по скорости метод Ньютона – Крылова, причем в некоторых случаях более чем в два раза.

Результаты вычислительных экспериментов
Results of the numerical experiments

n	Алгоритм			
	TSLs	TSLs + D	TSLs + WD	NK
Задача 1				
10 000	2,13/3636	0,84/1416	0,62/1016	0,84/680
22 500	10,59/8888	3,43/2832	1,54/1220	2,61/1050
40 000	34,22/16 968	8,71/4248	4,02/1829	6,57/1598
62 500	–	13,56/4248	7,89/2237	14,66/2341
90 000	–	26,86/5664	15,97/2951	25,25/2803
Задача 2				
10 000	14,25/4444	4,50/1416	3,23/1016	3,95/1031
22 500	67,17/9696	19,57/2832	9,91/1424	12,90/1564
40 000	208,17/16 463	36,40/2832	26,90/2033	39,85/2671
62 500	–	81,50/4248	47,90/2441	87,37/3901
90 000	–	157,31/5664	83,65/2951	148,77/4614
Задача 3				
10 000	2,43/4343	0,81/1416	0,69/1118	0,70/602
22 500	11,83/10 302	3,29/2832	1,62/1322	2,81/1176
40 000	36,61/18 685	5,64/2832	3,91/1829	7,94/1955
62 500	–	17,55/5664	7,21/2135	17,38/2816
90 000	–	25,69/5664	14,20/2747	33,20/3800

Дополнение. После принятия статьи к публикации авторами были обнаружены ранее опубликованные работы по сходной тематике [11; 12]. В англоязычной литературе описанный нами способ средне-квадратичного подавления ошибки называется ускорением Андерсона (Anderson acceleration), причем применяется это ускорение к обычному методу простой итерации. Новизна предложенного в настоящей работе подхода заключается в использовании в качестве базовой итерации двухшаговых процессов (6).

Библиографические ссылки

1. Faleichik B., Bondar I., Byl V. Generalized Picard iterations: A class of iterated Runge – Kutta methods for stiff problems // *J. Comput. Appl. Math.* 2014. Vol. 262. P. 37–50.
2. Saad Y. Iterative methods for Sparse Linear Systems. 2nd ed. Philadelphia : Siam, 2003.
3. Knoll D. A., Keyes D. E. Jacobian-free Newton – Krylov methods: a survey of approaches and applications // *J. Comput. Phys.* 2004. Vol. 193, issue 2. P. 357–397.
4. Фаддеев Д. К., Фаддеева В. Н. Вычислительные методы линейной алгебры. М. : Физматгиз, 1960.
5. Ортега Дж., Рейнболдт В. Итерационные методы решения нелинейных уравнений со многими неизвестными. М. : Мир, 1975.
6. Шанеев В. П., Ворожцов Е. В., Исаев В. И. и др. Метод коллокаций и наименьших невязок для трехмерных уравнений Навье – Стокса // *Вычисл. методы и программирование.* 2013. № 14. С. 306–322.
7. Курош А. Г. Курс высшей алгебры. 9-е изд. М. : Главная редакция физико-математической литературы, 1968.
8. Trefethen L. N., Bau D. III. Numerical Linear Algebra. Philadelphia : Siam, 1997.
9. Самарский А. А. Теория разностных схем. М. : Наука, 1977.
10. Baker A. H., Jessup E. R., Manteuffel T. A Technique for Accelerating the Convergence of Restarted GMRES // *SIAM J. Matrix Anal. Appl.* 2005. Vol. 26, issue 4. P. 962–984.
11. Walker H., Peng N. Anderson acceleration for fixed-point iterations // *SIAM J. Numer. Anal.* 2011. Vol. 49, issue 4. P. 1715–1735.
12. Toth A., Kelley C. T. Convergence Analysis for Anderson Acceleration // *SIAM J. Numer. Anal.* 2015. Vol. 53, issue 2. P. 805–819.

References

1. Faleichik B., Bondar I., Byl V. Generalized Picard iterations: A class of iterated Runge – Kutta methods for stiff problems. *J. Comput. Appl. Math.* 2014. Vol. 262. P. 37–50. DOI: 10.1016/j.cam.2013.10.036.
2. Saad Y. Iterative methods for Sparse Linear Systems. 2nd ed. Philadelphia : Siam, 2003.

3. Knoll D. A., Keyes D. E. Jacobian-free Newton – Krylov methods: a survey of approaches and applications. *J. Comput. Phys.* 2004. Vol. 193, issue 2. P. 357–397. DOI: 10.1016/j.jcp.2003.08.010.
4. Faddeev D. K., Faddeeva V. N. [Computational methods of linear algebra]. Moscow : Fizmatgiz, 1960 (in Russ.).
5. Ortega J., Rheinboldt V. [Iterative methods for solving nonlinear equations with many unknowns]. Moscow : Mir, 1975 (in Russ.).
6. Shapeev V. P., Vorozhtsov E. V., Isaev V. I., et al. [The method of collocations and least residuals for three-dimensional Navier – Stokes equations]. *Vychisl. metody program.* [Numer. Methods Program.]. 2013. No. 14. P. 306–322 (in Russ.).
7. Kurosh A. G. [The course of higher algebra]. 9th ed. Moscow : Glavnaya redaktsiya fiziko-matematicheskoi literatury, 1968 (in Russ.).
8. Trefethen L. N., Bau D. III. Numerical Linear Algebra. Philadelphia : Siam, 1997.
9. Samarsky A. A. [The theory of difference schemes]. Moscow : Nauka, 1977 (in Russ.).
10. Baker A. H., Jessup E. R., Manteuffel T. A Technique for Accelerating the Convergence of Restarted GMRES. *SIAM J. Matrix Anal. Appl.* 2005. Vol. 26, issue 4. P. 962–984. DOI: 10.1137/S0895479803422014.
11. Walker H., Peng N. Anderson acceleration for fixed-point iterations. *SIAM J. Numer. Anal.* 2011. Vol. 49, issue 4. P. 1715–1735. DOI: 10.1137/10078356X.
12. Toth A., Kelley C. T. Convergence Analysis for Anderson Acceleration. *SIAM J. Numer. Anal.* 2015. Vol. 53, issue 2. P. 805–819. DOI: 10.1137/130919398.

Статья поступила в редколлегию 21.03.2017.
Received by editorial board 21.03.2017.