УДК 519.688
DOI: 10.15827/0236-235X.118.270-281

Дата подачи статьи: 10.10.16
2017. Т. 30. № 2. С. 270–281

РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ НЬЮТОНОВСКИХ МЕТОДОВ БЕЗУСЛОВНОЙ ОПТИМИЗАЦИИ

А.Б. Свириденко, преподаватель, roshechka@gmail.com
(Новороссийский филиал Кубанского государственного университета,
ул. Героев Десантников, 87, г. Новороссийск, 353900, Россия)

Предложен подход к увеличению эффективности ньютоновских методов безусловной оптимизации, основанных на факторизации Холецкого, с регулировкой шага и с конечно-разностной аппроксимацией первых и вторых производных. В основе увеличения эффективности ньютоновских методов лежит модифицированное разложение Холецкого матрицы вторых производных, определяющее решение проблемы масштабирования шагов при спуске, аппроксимацию неквадратичными функциями, интеграцию с методом доверительной окрестности и уменьшение нормы априорной поправки. Исследована возможность уменьшения числа вычислений функции путем формирования матрицы вторых производных в соответствии с ее структурой.

Рассмотрена взаимосвязь подхода к увеличению эффективности гауссова исключения для разреженных матриц и предлагаемого подхода к увеличению эффективности численных методов ньютоновского типа – использование структуры матрицы, то есть информации о том, в каких позициях матрицы хранятся ненулевые элементы. Для ньютоновских методов безусловной оптимизации, основанных на факторизации Холецкого, с регулировкой шага и с конечно-разностной аппроксимацией первых и вторых производных это возможность уменьшения числа вычислений функции путем формирования матрицы вторых производных в соответствии с ее структурой.

Приведены описания программных реализаций, все версии алгоритмов реализованы на языке Visual Basic .NET, среда разработки – Microsoft Visual Studio 2010. Приведены результаты численного исследования эффективности реализованных алгоритмов с учетом ряда правил, описанных в работе.

Изучен подход к увеличению эффективности ньютоновских методов с конечно-разностной аппроксимацией первых и вторых производных. Подход является основой для дальнейших исследований, результаты которых могут быть использованы для построения численных методов ньютоновского типа.

Ключевые слова: ньютоновские методы, структура матрицы, конечно-разностная аппроксимация, факторизация Холецкого, масштабирование шагов, метод доверительной окрестности, норма априорной поправки, бенчмаркинг, профиль производительности, коллекция тестовых задач.

Ньютоновские методы – фундаментальные инструменты численного анализа, исследования операций оптимизации и управления. Например, большинство наиболее известных методов в линейном и нелинейном программировании строятся на их основе. Общий принцип построения большинства ньютоновских методов опишем следующим образом.

Рассмотрим задачу безусловной минимизации

$$\min_{x \in R^n} F(x), \quad (1)$$

где $F: R^n \rightarrow R^1$ – гладкая функция; R^n – n -мерное евклидово пространство.

Пусть h^k, H^k – градиент и гессиан, вычисленные на итерации k в точке x^k процесса решения задачи (1). Тогда общий принцип построения большинства ньютоновских методов безусловной оптимизации с регулировкой шага состоит в следующем. На каждой итерации сначала строится некоторая связанная с H^k существенно положительно определенная матрица $\bar{H}^k$, а затем направление спуска p^k вычисляется как решение системы $\bar{H}^k p^k = -h^k$ и определяется новая точка $x^{k+1} = x^k + \alpha_k p^k$, где α_k – длина шага, для которого $F^{k+1} < F^k$. При этом процедуру построения $\bar{H}^k$ организуют так, чтобы $\bar{H}^k$ совпадала с исходной матрицей H^k , если последняя сама является положительно определенной, причем выяснение определенности H^k и построение

$\bar{H}^k$ осуществляются параллельно в рамках одной процедуры на основе некоторых матричных разложений, которые позволяют выявить знаки собственных чисел H^k и приспособиться для генерации $\bar{H}^k$.

На практике условие гладкости целевой функции является излишне строгим: как правило, или первые и (или) вторые производные не существуют, или процесс вычисления этих производных слишком трудоемок. В этом случае быстро сходящийся алгоритм можно построить на основе их конечно-разностной аппроксимации. Подобно своему классическому прототипу он порождает целый класс модифицированных методов, причем модификации, предлагаемые для обобщения метода Ньютона на случаи со знаконеопределенными H^k , применимы и к его конечно-разностному аналогу, и наоборот [1, 2].

В данной работе в основе выяснения определенности H^k и построения $\bar{H}^k$ лежит факторизация (разложение) Холецкого. Разложение Холецкого для симметричной положительно определенной матрицы H^k имеет вид $H^k = L^k D^k (L^k)^T = (U^k)^T D^k U^k$, где $U^k = (L^k)^T$ – верхняя треугольная матрица с диагональными элементами, равными единице; D^k – диагональная матрица с положительными ведущими элементами по диагонали. Поскольку столбцы матрицы L^k с номерами с 1-го по $(j-1)$ -й

известны, ее j -й столбец определяется по формулам

$$d_{ii}^k = h_{ii}^k - \sum_{s=1}^{i-1} d_{is}^k (l_{js}^k)^2, \quad l_{ij}^k = \frac{1}{d_{ii}^k} \left(h_{ij}^k - \sum_{s=1}^{i-1} d_{is}^k l_{js}^k u_{is}^k \right).$$

Аналогичные формулы существуют и для построчной организации вычислений. В отличие от гауссовых исключений алгоритм Холецкого численно устойчив без каких-либо перестановок [3]. Это свойство определяется соотношением

$$\sum_{i=1}^n (l_{ij}^k)^2 d_{jj}^k = h_{jj}^k, \quad j = 1, \dots, n, \text{ между элементами } H^k \text{ и } L^k.$$

В силу него существует априорное ограничение сверху на элементы $r_{ij}^k = l_{ij}^k \sqrt{d_{jj}^k}$: каждый из них не превосходит максимальную величину h_{jj}^k . Соответственно лавинообразный рост элементов r_{ij}^k невозможен независимо от того, будут ведущие элементы малыми или нет.

Построение направлений ньютоновского поиска. Факторизация Холецкого неприменима для любой симметричной матрицы по следующим причинам. Во-первых, для знаконеопределенной матрицы H^k факторизация Холецкого может не существовать. Во-вторых, даже если она и существует, то гарантировать численную устойчивость алгоритма уже нельзя, поскольку никаких априорных ограничений на субдиагональные элементы L^k в рассматриваемом случае не будет. Итак, на основе обычной факторизации Холецкого построение $\bar{H}^k$ не получить, для этого нужно воспользоваться модифицированной факторизацией. Наиболее эффективная вычислительная схема предложена в [2, 3]. Данный подход состоит в том, чтобы строить факторы Холецкого D^k и L^k , подчиняющиеся двум требованиям: все диагональные элементы D^k должны быть существенно положительными, модули всех элементов треугольного фактора L^k должны быть равномерно ограничены сверху. Точнее говоря, требуется, чтобы для всех $j=1, 2, \dots, n$ и некоторых заданных положительных δ и β выполнялись неравенства

$$d_{jj}^k \geq \delta, \quad |r_{ij}^k| \leq \beta, \quad i > j, \quad (2)$$

где r_{ij}^k – введенные для удобства изложения вспомогательные величины, по определению равные $u_{ij}^k \sqrt{d_{jj}^k}$. Гринштадт предложил в [4] при реализации алгоритма на конкретной ЭВМ, в которой под запись мантииссы отводится t битов, величину δ вычислять по формуле $\delta = \max \{ 2^{-t} \|H^k\|_{\infty}, 2^{-t} \}$.

Для сохранения численной устойчивости процедуры построения H^k , а также для совпадения H^k и $\bar{H}^k$ в случае положительно определенной H^k целесообразно величину β вычислять по формуле $\beta^2 = \max \left\{ \gamma, \xi / \sqrt{n^2 - 1}, \varepsilon_M \right\}$.

Здесь ξ – максимальный модуль недиагонального элемента H^k ; γ – значение максимального из диагональных элементов H^k ; ε_M – машинная точность; ε_M вводится в формулу расчета β , чтобы обеспечить устойчивость вычислений, когда норма H^k очень мала [4]. При этом процедура расчета модифицированных факторов L^k и D^k фактически представляет собой обычный алгоритм факторизации Холецкого с попутным увеличением (по мере необходимости) диагонали исходной матрицы с целью добиться выполнения неравенств (2). Матрицы L^k и D^k , полученные по окончании описанной процедуры, будут факторами Холецкого для положительно определенной матрицы $\bar{H}^k$, связанной с H^k следующим образом: $L^k D^k (L^k)^T = H^k + E^k + \bar{H}^k$, где E^k – неотрицательная диагональная матрица, j -й элемент которой равен e_{jj}^k . Таким образом, положительно определенная матрица $\bar{H}^k$ может отличаться от исходной матрицы H^k только диагональными элементами.

Направление спуска p^k вычисляется как решение системы $\bar{H}^k p^k = -h^k$, причем $p^k = -(\bar{H}^k)^{-1} h^k$

определяют последовательным решением двух систем линейных уравнений с треугольными матрицами: $L^k y^k = -h^k$, $D^k (L^k)^T p^k = y^k$.

Для построения модифицированного разложения Холецкого требуется выполнить около $\frac{1}{6} n^3$ арифметических операций, примерно столько же, сколько необходимо для обычного разложения для положительно определенной матрицы.

Вычислительная схема 1 модифицированного разложения Холецкого. Далее приводится детальное описание всех операций, выполняемых по ходу построения модифицированного разложения Холецкого с перестановками [3]. Дана наиболее эффективная схема организации расчетов. В процессе вычисления j -го столбца матрицы L^k участвуют вспомогательные величины $c_{is}^k = l_{is}^k d_{ss}^k$, $s = 1, \dots, j$, $i = j, \dots, n$.

Шаг 0 (Расчет порога для элементов). Вычислить $\beta^2 = \max \{ \gamma^0, \xi / v, \varepsilon_M \}$, где $v = \max \{ 1, \sqrt{n^2 - 1} \}$, а числа γ и ξ – максимальные значения модулей диагонального и недиагонального элементов H^k .

Шаг 1 (Инициализация). Присвоить индексу столбца j значение 1. Принять $c_{ii}^k = g_{ii}^k$, $i = 1, \dots, n$.

Шаг 2 (Перестановка строк и столбцов). Найти индекс q , такой, что $|c_{qq}^k| = \max_{j \leq i \leq n} |c_{ii}^k|$. Поменять местами все данные, отвечающие столбцам матрицы H^k с номерами q и j , а затем проделать то же самое с данными, отвечающими ее q -й и j -й строкам.

Шаг 3 (Поиск максимальной по модулю величины $l_{ij}^k d_{ii}^k$). Вычислить $c_{ij}^k = g_{ij}^k - \sum_{s=1}^{j-1} l_{js}^k c_{is}^k$ для

$i = j+1, \dots, n$ и найти $\theta_j = \max_{j+1 \leq i \leq n} |c_{ij}^k|$ (если $j = n$, взять $\theta_j = 0$).

Шаг 4 (Расчет j -го диагонального элемента фактора D^k). Вычислить $d_{jj}^k = \max \{ \delta, |c_{jj}^k|, \theta_j^2 / \beta^2 \}$ и поправку $e_{jj}^k = d_{jj}^k - c_{jj}^k$. Если $j = n$, вычисления прекратить.

Шаг 5 (Расчет j -й строки L^k). Вычислить $l_{js}^k = c_{js}^k / d_{ss}^k$ для $s = 1, \dots, j-1$. Для $i = j+1, \dots, n$ пересчитать диагональные элементы c_{ii}^k по формуле $c_{ii}^k = c_{ii}^k - l_{ij}^k c_{ij}^k$. Присвоить j значение $j+1$ и вернуться к шагу 2.

Модифицированная факторизация Холецкого, помимо прочего, позволяет определить и направление отрицательной кривизны, решая уравнение $(L^k)^T p^k = e_s$, где индекс s выбирается из условия $d_{ss}^k - e_{ss}^k \leq d_{jj}^k - e_{jj}^k$, $j = 1, \dots, n$.

Подход к увеличению эффективности алгоритма Гилла и Мюррея предложен в [1]. Доказано, что подход к построению направления спуска p^k определяет решение проблемы масштабирования шагов при спуске и интеграцию с методом доверительной окрестности следующим образом. В алгоритме Гилла и Мюррея направление спуска p^k в конечном счете вычисляется как решение системы

$$(L^k)^T p^k = U^k p^k = u^k, \quad (3)$$

где $u^k = (D^k)^{-1} y^k$, а y^k – это решение системы $L^k y^k = -h^k$. Величина элементов $|u_j^k|$ зависит от способа задания направления спуска. Следуя подходу Гилла и Мюррея, для численной устойчивости расчета элементов D^k , L^k , u^k достаточно потребовать изменения способа задания p^k так, чтобы для всех $j = 1, 2, \dots, n$ и некоторых заданных положительных δ , β и ω выполнялись неравенства

$$d_{jj}^k \geq \delta, |r_{ij}^k| \leq \beta, i > j, |u_j^k| \leq \omega.$$

Это может стать ключом к решению проблемы масштабирования шагов при спуске, но такое предположение ошибочно: подход линейной алгебры к вычислению направления спуска p^k исключает расчет элементов вектора u^k по ходу построения модифицированного разложения Холецкого [5, 6]. Нужен альтернативный подход, который опишем следующим образом.

Равенства $H^k = (U^k)^T D^k U^k$ достаточно для определения элементов матриц D^k , U^k . В скалярной форме это равенство выглядит следующим образом: $h_{ij}^k = \sum_{\mu=1}^i u_{\mu i}^k u_{\mu j}^k d_{\mu\mu}^k$. Отсюда, полагая $u_{ii}^k = 1$, построим соотношения

$$d_{ii}^k = h_{ii}^k - \sum_{s=1}^{i-1} (u_{si}^k)^2 d_{ss}^k, \quad (4)$$

$$u_{ij}^k = \left(h_{ij}^k - \sum_{s=1}^{i-1} u_{si}^k u_{sj}^k d_{ss}^k \right) / d_{ii}^k, j > i,$$

для расчета элементов D^k , U^k . Построение факторов Холецкого математически эквивалентно применению метода исключения Гаусса к системе уравнений $H^k p^k = -h^k$ в прямом порядке, при этом техника исключения Гаусса позволяет получить соотношение

$$u_i^k = - \left(h_i^k - \sum_{s=1}^{i-1} u_s^k u_{si}^k d_{ss}^k \right) / d_{ii}^k \quad (5)$$

для расчета элементов вектора u^k и систему уравнений (3) для вычисления направления спуска p^k . Интеграция техники исключения Гаусса и факторизации Холецкого приводит к соотношениям (4) и (5), которые с помощью элементарных арифметических операций позволяют вычислить элементы u^k по ходу построения факторов Холецкого.

Интеграция техники исключения Гаусса и факторизации Холецкого генерирует множество реализаций численно устойчивых вычислительных схем задания направления спуска p^k , в основе которых лежит требование, чтобы на очередном шаге вычислений коэффициентов факторов Холецкого соответствующий диагональный элемент матрицы D^k и соответствующий элемент вектора u^k сначала рассчитывались по вычисленным ранее значениям этих коэффициентов. Затем диагональный элемент D^k увеличивается настолько, насколько необходимо, чтобы все диагональные элементы D^k были существенно положительными, модули всех элементов U^k и u^k были равномерно ограничены сверху. Например, достаточно потребовать, чтобы для всех $i = 1, 2, \dots, n$ выполнялись неравенства

$$d_{ii}^k \geq \delta, |u_{ij}^k| \leq 1, j > i, |u_j^k| \leq 1, \quad (6)$$

положив $d_{ii}^k = \max \{ \delta, |c_{ii}^k|, \theta_i^2 / \beta^2, |c_i^k|, \theta_i \}$, поскольку элементы d_{ii}^k , c_{ij}^k / d_{ii}^k и c_i^k / d_{ii}^k определяют i -е строки матриц D^k , U^k , u^k и удовлетворяют неравенствам (6). Здесь c_{ii}^k , c_{ij}^k , c_i^k – вспомогательные величины, определяемые следующим образом:

$$c_{ii}^k = h_{ii}^k - \sum_{s=1}^{i-1} (u_{si}^k)^2 d_{ss}^k,$$

$$c_{ij}^k = h_{ij}^k - \sum_{s=1}^{i-1} u_{si}^k u_{sj}^k d_{ss}^k, j > i, \quad (7)$$

$$c_i^k = - \left(h_i^k - \sum_{s=1}^{i-1} u_s^k u_{si}^k d_{ss}^k \right).$$

Интеграция техники исключения Гаусса и факторизации Холецкого генерирует множество реализаций вычислительных схем интеграции с методом доверительной окрестности, например, следующим образом. В процессе построения D^k , U^k , u^k вычислить элементы $g_i^k = d_{ii}^k / l_i^k$ (если $d_{ii}^k > l_i^k \gamma$, взять $g_i^k = l_i^k / \gamma$), где $l_i^k = \max \{ \delta, |d_{ii}^k| \}$, вычислить $\gamma^k = \max_{1 \leq i \leq n} g_i^k$ и определить направление спуска p^k из системы

$$\frac{1}{\gamma^k} \bar{H}^k p^k = -h^k. \quad (8)$$

Здесь параметр γ задается пользователем, число γ^k ($\gamma^k \geq 1$ по построению) характеризует степень однородности, при этом (8) – простейшая форма аппроксимации, отличная от квадратичной. Пусть $\bar{H}^k p^k = -\gamma^k h^k = -h^k - (\gamma^k - 1)h^k = -h^k - G^k p^k$, где $G^k -$ диагональная матрица с элементами $g_{ii}^k = \frac{(\gamma^k - 1)h_i^k}{p_i^k}$

на диагонали. Тогда $(\bar{H}^k + G^k) p^k = -h^k$, что означает сдвиг на g_{ii}^k всех i -х собственных значений матрицы $\bar{H}^k$. В методах с регулировкой шага матрицу H^k модифицируют так, чтобы изменения не затрагивали подпространства, натянутого на ее собственные векторы с положительными собственными значениями [3]. Если же замена осуществляется в методе доверительной окрестности, она отражается на всех векторах, так как результатом замены H^k на матрицу $\bar{H}^k = H^k + \lambda^k I$ будет сдвиг на λ^k всех собственных значений матрицы H^k [3]. Из вышесказанного следует, что стратегия выбора направления спуска определяет и интеграцию с методом доверительной окрестности.

Вычислительная схема 2 модифицированного разложения Холесского. Далее приведена наиболее эффективная схема организации расчетов. Все фигурирующие в ней величины при реализации на ЭВМ могут размещаться в памяти, первоначально выделяемой для записи матриц H^k и h^k . При этом коэффициенты рассчитываемых факторов занимают места ее использованных элементов. В процессе вычисления i -й строки фактора U^k участвуют вспомогательные величины (7). В процессе построения системы уравнений $U^k p^k = u^k$ на каждой итерации k выявляются знаки собственных чисел и вычисляется количество отрицательных n_o и нулевых n_0 собственных значений, число переходов к направлению отрицательной кривизны n_S .

Шаг 0. Вычислить $\beta^2 = \max \{ \gamma^0, \xi / \nu, \varepsilon_M \}$, где $\nu = \max \{ 1, \sqrt{n^2 - 1} \}$, а числа γ^0 и ξ – максимальные значения модулей диагонального и недиагонального элементов H^k . Положить $\varepsilon_0 = 2^{-\frac{\tau_F}{2}}$, $\varepsilon_s = 2^{-\frac{\tau_F}{3}} (1 + \|F(x^{k-1}) - F(x^k)\|)$. Здесь τ_F – число правильных разрядов $F(x^k)$, которые хотелось бы получить.

Шаг 1. Принять $i = 1$ и $c_{jj}^k = h_{jj}^k$ ($j = 1, \dots, n$), $c_j^k = -h_j^k$ ($j = 1, \dots, n$).

Шаг 2. Найти такой индекс q , что $|c_{qq}^k| = \max_{i \leq j \leq n} |c_{ij}^k|$, и поменять местами все данные, отвечающие строкам матрицы H^k с номерами q и i , а затем проделать то же самое с данными, отвечающими ее q -му и i -му столбцам.

Шаг 3. Если $c_{ii}^k < \varepsilon_0$, то положить $l_i^k = \delta$; иначе положить $l_i^k = c_{ii}^k$.

Шаг 4. Если $|c_{ii}^k| \leq \varepsilon_0$, то положить $n_0 = n_0 + 1$ и перейти к шагу 8; иначе перейти к шагу 5.

Шаг 5. Если $c_{ii}^k > 0$, то перейти к шагу 8; иначе перейти к шагу 6.

Шаг 6. Принять $n_0 = n_0 + 1$. Если $\|h^k\| > \varepsilon_s$, то перейти к шагу 8; иначе перейти к шагу 7.

Шаг 7. Построить систему уравнений $U^k p^k = e_i$ для поиска направления отрицательной кривизны по формулам $u_j^k = 0$, $j = 1, \dots, n$, $u_i^k = 1$, положить $n_S = n_S + 1$ и остановиться.

Шаг 8. Если $i = n$, то положить $\theta_i = 0$; иначе вычислить вспомогательные величины c_{ij}^k , $j = i + 1, \dots, n$, по формуле $c_{ij}^k = h_{ij}^k - \sum_{s=1}^{i-1} u_{si}^k u_{sj}^k d_{ss}^k$ и найти

$$\theta_i = \max_{i+1 \leq j \leq n} |c_{ij}^k|.$$

Шаг 9. Вычислить диагональный элемент фактора D^k и элемент вектора u^k по формулам $d_{ii}^k = \max \{ \delta, |c_{ii}^k|, \theta_i^2 / \beta^2, |c_i^k|, \theta_i \}$, $u_i^k = c_i^k / d_{ii}^k$. Вычислить $g_i^k = d_{ii}^k / l_i^k$ (если $d_{ii}^k > l_i^k \gamma$, взять $g_i^k = l_i^k / \gamma$). Если $i = n$, то перейти к шагу 11.

Шаг 10. Вычислить строку $u_{ij}^k = c_{ij}^k / d_{ii}^k$, $j = i + 1, \dots, n$, фактора U^k , пересчитать вспомогательные величины по формулам $c_{jj}^k = c_{jj}^k - u_{ij}^k c_{ij}^k$, $j = i + 1, \dots, n$, $c_j^k = c_j^k - u_{ij}^k c_i^k$, $j = i + 1, \dots, n$, положить $i = i + 1$ и перейти к шагу 2.

Шаг 11. Найти $\gamma^k = \max_{1 \leq i \leq n} g_i^k$, положить $u^k = \gamma^k u^k$ и остановиться.

Замечания. Предложенный подход к вычислению длины шага можно найти в [7–9]. В основе лежит интеграция подходов [3, 10–12].

Предложенный подход к модификации критериев останова, разработанных Гиллом, Мюрреем и Райт [3], можно найти в [6].

Предложенный подход к оцениванию конечно-разностных интервалов и подход к конечно-разностной аппроксимации первых и вторых производных можно найти в [1]. В основе лежит интеграция подходов [3, 11], там же описаны способы выбора конечно-разностных интервалов. Стандартные программы численного дифференцирования нацелены на поиск конечно-разностного интервала, обеспечивающего минимальную суммарную ошибку вычисляемого приближения. Алгоритмы выбора интервала для центральной конечно-разностной формулы можно найти, например, в [13]. Однако для оптимизационных приложений такие программы не подходят, так как, обеспечивая избыточную точность вычисления

оценок производных, они обычно требуют слишком большого числа обращений к процедуре расчета значений функции.

В [14] отмечено, что замена градиентов их конечно-разностными оценками лучше всего проходит в квазиньютоновских методах. Для методов ньютоновского типа результаты получаются менее удовлетворительными.

Тестирование

Построение алгоритмов, в том числе и развитие подхода Гилла и Мюррея к построению направления спуска [1, 6, 9, 14], требует постоянного тестирования на больших классах тестовых задач различных структур и характеристик с различными настройками с целью выявления слабых мест – ситуаций, приводящих к замедлению алгоритма или ухудшению качества результата и т.д.

Алгоритмы безусловной оптимизации должны быть протестированы по крайней мере в двух различных смыслах:

- для оценки конкретной реализации и соответствующих вычислительных процедур; очевидно, что хорошо разработанные тестовые задачи являются очень мощным инструментом для уточнения концепций и механизмов алгоритма;
- для получения представления о гипотезе, доказательстве работоспособности алгоритма и сравнения алгоритмов на экспериментальном уровне.

Как правило, задачи безусловной оптимизации можно разделить на два типа: искусственные и реальные. Искусственные задачи используются для того, чтобы увидеть поведение алгоритма в различных трудных ситуациях, таких как оптимизация функций с длинными узкими оврагами, со значительными нуль-пространственными эффектами, унимодальных, с огромным числом существенных локальных экстремумов и т.д. Основной характеристикой искусственных задач является то, что их относительно легко настраивать и применять в процессе построения алгоритма.

Реальные задачи берутся из разных источников прикладных оптимизационных задач физики, химии, техники, биологии, экономики, океанологии, астрономии, метеорологии и т.д. В отличие от искусственных настройки реальных задач труднодоступны и ими нелегко оперировать. Они могут иметь сложные алгебраические или дифференциальные выражения, могут зависеть от огромного количества данных и, возможно, зависеть от некоторых параметров, которые должны быть оценены определенным образом. Очень хорошая коллекция реальных задач безусловной оптимизации описана в работах [15, 16]. Следует отметить, что собрание коллекций реальных задач началось в конце 1970-х годов с отстаивания высокого качества отчетности о вычислительных экспериментах математического программирования с ПО [17–19].

Источником тестовых задач, рассмотренных в данной работе, является коллекция [20], в состав которой вошли тестовые функции из коллекции CUTE и прочих [21–24], а также из других работ и технических отчетов. Цель этой коллекции – представить большое количество общих тестовых функций, которые могут использоваться для тестирования алгоритмов безусловной оптимизации и сравнения исследований. Для каждой функции даны алгебраические представления и начальные (стартовые) точки. Для всех задач из коллекции [20] известные алгоритмы безусловной оптимизации, например, квазиньютоновские, сходятся к одному и тому же локальному решению.

Опишем конкретные тестовые задачи [20].

Задача 1 (Extended Rosenbrock function):

$$F(x) = \sum_{i=1}^{n/2} 100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2,$$

$$x^0 = (-1.2 \ 1 \ \dots \ -1.2 \ 1)^T.$$

Задача 2 (Wood function):

$$F(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1),$$

$$x^0 = (-3 \ -1 \ -3 \ -1)^T.$$

Задача 3 (Generalized Rosenbrock function):

$$F(x) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2,$$

$$x^0 = (-1.2 \ 1 \ \dots \ -1.2 \ 1)^T.$$

Задача 4 (Extended White & Holst function):

$$F(x) = \sum_{i=1}^{n/2} 100(x_{2i} - x_{2i-1}^3)^2 + (1 - x_{2i-1})^2,$$

$$x^0 = (-1.2 \ 1 \ \dots \ -1.2 \ 1)^T.$$

Задача 5 (Extended Penalty function):

$$F(x) = \sum_{i=1}^{n-1} 100(x_i - 1)^2 + \left(\sum_{j=1}^n x_j^2 - 0.25 \right)^2,$$

$$x^0 = (1 \ 2 \ \dots \ n)^T.$$

Задача 6 (Perturbed Quadratic function):

$$F(x) = \sum_{i=1}^n ix_i^2 + \frac{1}{100} \left(\sum_{i=1}^n x_i \right)^2,$$

$$x^0 = (0.5 \ 0.5 \ \dots \ 0.5)^T.$$

Задача 7 (Raydan 1 function):

$$F(x) = \sum_{i=1}^n \frac{i}{10} (\exp(x_i) - x_i), \quad x^0 = (1 \ 1 \ \dots \ 1)^T.$$

Задача 8 (Raydan 2 function):

$$F(x) = \sum_{i=1}^n (\exp(x_i) - x_i), \quad x^0 = (1 \ 1 \ \dots \ 1)^T.$$

Задача 9 (Diagonal 1 function):

$$F(x) = \sum_{i=1}^n (\exp(x_i) - ix_i), \quad x^0 = (1/n \ 1/n \ \dots \ 1/n)^T.$$

Задача 10 (Diagonal 2 function):

$$F(x) = \sum_{i=1}^n \left(\exp(x_i) - \frac{x_i}{i} \right), \quad x^0 = (1/1 \ 1/2 \ \dots \ 1/n)^T.$$

Задача 11 (Diagonal 3 function):

$$F(x) = \sum_{i=1}^n (\exp(x_i) - i \sin(x_i)), \quad x^0 = (1 \ 1 \ \dots \ 1)^T.$$

Завершение итераций производилось с использованием рекомендаций из [6]:

$$|F(x^{k-1}) - F(x^k)| < 2^{-\tau_F} (1 + |F(x^{k-1}) - F(x^k)|),$$

$$\|h^k\| \leq 2^{-\frac{\tau_F}{3}} (1 + |F(x^{k-1}) - F(x^k)|),$$

$$\|h^k\| \leq 2^{-\frac{\tau_F}{3}} (1 + |F(x^k)|),$$

что является ужесточением условий останова из [4], позволяющим повысить точность решения, и хорошо согласуется с численными экспериментами [1].

Замечания. Трудности сравнения нескольких методов оптимизации по результатам численных экспериментов состоят в следующем.

Во-первых, сравнению подвергаются не методы, а машинные реализации соответствующих алгоритмов. Результат сравнения зависит от качества программного кода, выбора параметров алгоритма, числовых типов данных и т.д.

Во-вторых, неясно, каким образом соизмерять трудоемкость различных методов. Естественный, на первый взгляд, критерий – затраченное машинное время – в действительности не подходит: сложно сравнивать быстродействие различных машин, не всегда имеются данные о затратах времени при работе в мультипроцессорном режиме и т.д. Более надежным показателем является число вычислений минимизируемой функции или другая внутренняя характеристика метода. Но и здесь возникает ряд проблем. Например, неясно, как соизмерять трудоемкость вычисления функции и решения различных вспомогательных задач. Затруднительно сопоставлять вычисления функции и ее производных. В тех задачах, где применяются конечно-разностные аппроксимации производных, все намного проще: одно вычисление градиента эквивалентно n вычислениям функции, вычисление гессиана эквивалентно $n(n+1)/2$ вычислениям функции. Однако для дискретной задачи оптимального управления градиент лишь примерно вдвое «дороже» функции [25]. Для квадратичной функции градиент вычисляется даже проще, чем ее значения. Так же дело обстоит и с субградиентом в минимаксной задаче и т.д.

В-третьих, методы могут по-разному вести себя на разных этапах процесса минимизации. Более того, поведение разных величин, характеризующих точность решения, может быть также различным.

Никакого удовлетворительного способа преодоления перечисленных трудностей не существует. Единственное, что можно сделать в подобной ситуации, – приводить данные о результатах вычислений в развернутой форме, чтобы иметь возможность сравнения методов по разным критериям. Рекомендуется при публикации результатов

проверки методов придерживаться следующих правил [25]:

- приводить точную формулировку задачи, для которой проводился счет, включая все ее параметры и начальное приближение;
- указывать тип ЭВМ и данных, язык и среду программирования, сведения о программе;
- давать подробное описание применяемого алгоритма или отсылать к его публикации, если она имеется;
- сообщать различные характеристики точности приближения, невязку в ограничениях и выполнении условий экстремума; в задачах малой размерности приводить и сами приближения;
- указывать подробные сведения о трудоемкости вычислений.

При выполнении этих условий можно воспроизводить полученные данные и сравнивать их с другими по разным показателям.

Далее x^* – точка глобального минимума $F(x)$ на R^n , $F^* = F(x^*)$; k – число итераций;

$\delta F = |F(x^*) - F(x^k)|$ – точность решения по функции; $\delta x = \max |x_i^* - x_i^k|$ – точность решения по аргументам; k_0 – число вычислений функции; k_1 – число вычислений функции (регулировка шага); $\nabla F(x)$ – градиент $F(x)$; $\nabla^2 F(x)$ – матрица вторых производных (гессиан).

Тестовые задачи с известным решением можно построить следующим образом. Выбираются точка x^* и функции $\varphi_i(x)$, $i = 1, 2, \dots, m$. Тогда

$$F(x) = \sum_{i=1}^m (\varphi_i(x) - \varphi_i(x^*))^2 \quad (9)$$

достигает минимума в точке x^* . При этом $F^* = 0$,

$\nabla^2 F(x^*) = \sum_{i=1}^m \nabla \varphi_i(x^*) \nabla^T \varphi_i(x^*)$, так что можно регулировать обусловленность задачи за счет выбора $\varphi_i(x)$. В частности, если $m < n$ или $m = n$, но $\nabla \varphi_i(x^*) = 0$ для некоторых i , получаем вырожденную точку минимума. Функции вида (9), вообще говоря, невыпуклы и могут иметь локальные и даже глобальные минимумы, отличные от x^* .

Другой способ связан с выбором функций вида

$$F(x) = \sum_{i=1}^{n+1} f_i(x_i, x_{i-1}), \quad x_0 = a, \quad x_{n+1} = b. \quad (10)$$

Тогда, задавшись $x_0^* = a$ и решив последовательно одномерные уравнения

$$\partial F / \partial x_i = \partial / \partial x_i (f_i(x_i^*, x_{i-1}^*) + f_{i+1}(x_{i+1}, x_i^*)) = 0 \quad (11)$$

(решение обозначим x_{i+1}^*), для $i = 0, 1, \dots, n$ выберем $b = x_{n+1}^*$. Очевидно, что тогда $\nabla F(x^*) = 0$. Если $f_i(x_i, x_{i-1})$ выпуклы по x_i , x_{i-1} , то $F(x)$ выпукла и x^* – точка минимума $F(x)$.

Наконец, можно взять произвольную гладкую выпуклую функцию $\varphi(x)$ и произвольную точку x^*

и построить $F(x) = \varphi(x) - (\nabla \varphi(x^*), x)$. Тогда $\nabla F(x^*) = 0$, $F(x)$ выпукла и потому x^* – точка глобального минимума $F(x)$. Существуют и другие приемы построения функций с известной точкой минимума.

Следует отметить, что для функций вида (9), (10) существуют специальные методы, обладающие повышенной эффективностью. Например, для задач (11) эффективен метод Гаусса–Ньютона [3], для (11) – метод динамического программирования и покоординатного спуска [25]. Аналогичным образом специфика задач (9), (11) может сказаться на поведении общих методов оптимизации.

Развитие подхода и численные исследования

В [3] доказано, что значение нормы поправки E^k можно дополнительно уменьшить, если использовать симметричные перестановки строк и столбцов H^k на шаге 2 вычислительной схемы 1 модифицированного разложения Холецкого следующим образом. Найти такой индекс q , что $|c_{qq}^k| = \max_{i \leq j \leq n} |c_{ij}^k|$, и поменять местами все данные, отвечающие строкам матрицы H^k с номерами q и i , а затем проделать то же самое с данными, отвечающими ее q -му и i -му столбцам.

Подход Гилла и Мюррея к дополнительному уменьшению нормы поправки E^k развит в [6]. Доказано, что фактическое значение нормы поправки E^k можно дополнительно уменьшить, если использовать симметричные перестановки строк и столбцов H^k . На очередном, i -м, шаге факторизации в качестве i -й строки и i -го столбца надо брать ту из нетронутых пар, для которой величина $|c_{ii}^k| + |c_i^k|$ максимальна.

Такая стратегия дает возможность увеличить численную устойчивость расчета элементов D^k , U^k , u^k и остается работоспособной и в том случае, когда $\|h^k\|$ не равна нулю, но очень мала. Там же отмечено: «Параметр β подбирается путем минимизации априорной оценки нормы поправки E^k и с очевидностью зависит от способа задания направления спуска и, как следствие, от величин c_{ij}^k и g_i^k . Рассмотрение предлагаемой формулы вычисления величины β , учитывающей такую зависимость, не является целью данной работы и находится в стадии экспериментального сравнения эффективности». В данной работе величину β предлагается вычислять по формуле

$$\beta^2 = \max \{ \gamma^0, \rho, \xi / \nu, \varepsilon_M \}, \quad (12)$$

где $\rho = \max_{1 \leq i \leq n} |h_i^k|$, ε_M – машинная точность; ε_M вводится в формулу расчета β , чтобы обеспечить устойчивость вычислений, когда норма H^k очень мала. Применение формулы (12) хорошо согласо-

уется с численными экспериментами, описанными в данной работе. Ниже приводится предлагаемая модификация вычислительной схемы 2, дана наиболее эффективная схема организации расчетов.

Вычислительная схема 3 модифицированного разложения Холецкого. Все фигурирующие в ней величины при реализации на ЭВМ могут размещаться в памяти, первоначально выделяемой для записи матриц H^k и h^k .

Шаг 0. Вычислить $\beta^2 = \max \{ \gamma^0, \rho, \xi / \nu, \varepsilon_M \}$,

где $\rho = \max_{1 \leq i \leq n} |h_i^k|$, $\nu = \max \{ 1, \sqrt{n^2 - 1} \}$, а числа γ^0

и ξ – максимальные значения модулей диагонального и недиагонального элементов H^k , ε_M –

машинная точность. Положить $\varepsilon_0 = 2^{-\frac{\tau_F}{2}}$,

$\varepsilon_s = 2^{-\frac{\tau_F}{3}} \left(1 + \|F(x^{k-1}) - F(x^k)\| \right)$. Здесь τ_F – число правильных разрядов $F(x^k)$, которые хотелось бы получить.

Шаг 1. Положить $i = 1$ и $c_{jj}^k = h_{jj}^k$ ($j = 1, \dots, n$), $c_j^k = -h_j^k$ ($j = 1, \dots, n$).

Шаг 2. Найти индекс q , такой, что $|c_{qq}^k| = \max_{i \leq j \leq n} (|c_{ij}^k| + |c_j^k|)$, и поменять местами все дан-

ные, отвечающие строкам матрицы H^k с номерами q и i , а затем проделать то же самое с данными, отвечающими ее q -му и i -му столбцам.

Шаг 3. Если $c_{ii}^k < \varepsilon_0$, то положить $l_i^k = \delta$; иначе положить $l_i^k = c_{ii}^k$.

Шаг 4. Если $|c_{ii}^k| \leq \varepsilon_0$, то положить $n_0 = n_0 + 1$ и перейти к шагу 8; иначе перейти к шагу 5.

Шаг 5. Если $c_{ii}^k > 0$, то перейти к шагу 8; иначе перейти к шагу 6.

Шаг 6. Положить $n_0 = n_0 + 1$. Если $\|h^k\| > \varepsilon_s$, перейти к шагу 8; иначе перейти к шагу 7.

Шаг 7. Построить систему уравнений $U^k p^k = e_i$ для поиска направления отрицательной кривизны по формулам $u_j^k = 0$, $j = 1, \dots, n$, $u_i^k = 1$, положить $n_s = n_s + 1$ и остановиться.

Шаг 8. Если $i = n$, то положить $\theta_i = 0$; иначе вычислить вспомогательные величины c_{ij}^k , $j = 1, \dots, n$,

по формуле $c_{ij}^k = h_{ij}^k - \sum_{s=1}^{i-1} u_{is}^k u_{sj}^k d_{ss}^k$ и найти

$$\theta_i = \max_{i+1 \leq j \leq n} |c_{ij}^k|.$$

Шаг 9. Вычислить диагональный элемент фактора D^k и элемент вектора u^k по формулам $d_{ii}^k = \max \{ \delta, |c_{ii}^k|, \theta_i^2 / \beta^2, |c_i^k|, \theta_i \}$, $u_i^k = c_i^k / d_{ii}^k$. Вычислить $g_i^k = d_{ii}^k / l_i^k$ (если $d_{ii}^k > l_i^k \gamma$, взять $g_i^k = l_i^k / \gamma$). Если $i = n$, перейти к шагу 11.

Шаг 10. Вычислить строку $u_{ij}^k = c_{ij}^k / d_{ij}^k$, $j = 1, \dots, n$, фактора U^k , пересчитать вспомогательные величины по формулам $c_{jj}^k = c_{jj}^k - u_{ij}^k c_{ij}^k$, $j = i+1, \dots, n$, $c_j^k = c_j^k - u_{ij}^k c_i^k$, $j = i+1, \dots, n$, положить $i = i+1$ и перейти к шагу 2.

Шаг 11. Найти $\gamma^k = \max_{1 \leq i \leq n} g_i^k$, положить $uk = \gamma^k u^k$ и остановиться.

В данной работе отмечена взаимосвязь подхода к увеличению эффективности гауссова исключения для разреженных матриц в [26] и предлагаемого подхода к увеличению эффективности численных методов ньютоновского и квазиньютоновского типов – использование структуры матрицы, то есть информации о том, в каких позициях матрицы хранятся ненулевые элементы. Для ньютоновских и квазиньютоновских методов безусловной оптимизации, основанных на факторизации Холецкого, с регулировкой шага и с конечно-разностной аппроксимацией первых и вторых производных – это возможность уменьшения числа вычислений функции путем формирования матрицы вторых производных в соответствии с ее структурой. Данное исследование является прямым продолжением работы [14].

Программные реализации. Приложения Nmbm и NmbmApp – машинная реализация ньютоновских методов оптимизации с регулировкой шага, основанных на факторизации Холецкого (вычислительная схема 2 модифицированного разложения Холецкого) [27, 28].

NmbmApp отличается от своего классического прототипа Nmbm конечно-разностной аппроксимацией первых и вторых производных. Все версии алгоритмов реализованы на языке Visual Basic .NET, среда разработки – Microsoft Visual Studio 2010. Исходные тексты программ доступны на сайте <http://crm.ics.org.ru> и могут быть использованы по универсальной общедоступной лицензии GNU.

Приложение NmbmsApp отличается от своего классического прототипа NmbmApp формированием матрицы вторых производных в соответствии с ее структурой. NmbmsApp – машинная реализация ньютоновских методов оптимизации с регулировкой шага, основанных на факторизации Холецкого (вычислительная схема 3 модифицированного разложения Холецкого).

Главное окно программы и пример результатов решения одной из тестовых задач размещены по ссылке http://www.swsys.ru/uploaded/image/2017_2/2017-2-dop/6.jpg/.

Достоинством подхода, основанного на применении формул численного дифференцирования, кроме его универсальности, является низкая стоимость подготовки задачи к компьютерному моделированию [29]. От пользователя требуется лишь написание программы для вычисления значения $F(x)$ при заданном x , поэтому реализованные на ос-

нове численных производных методы оптимизации оказываются по существу прямыми методами, то есть методами, не использующими в своей схеме производные от $F(x)$. В то время как результаты численных исследований, приведенные в [14], показывают, что конечно-разностные аналоги проигрывают в точности решения следующих задач оптимизации.

Задача 12 (функция Поляка, среднеквадратичная аппроксимация экспонентами):

$$F(x) = \sum_{j=1}^{10} [\exp(-0.2j) + 2 \exp(-0.4j) - x_1 \exp(-0.2jx_2) - x_3 \exp(-0.4jx_4) - x_1 \exp(-0.2jx_2) - x_3 \exp(-0.4jx_4)]^2,$$

$$F(x) = \sum_{j=1}^{10} [\exp(-0.2j) + 2 \exp(-0.4j) - x_1 \exp(-0.2jx_2) - x_3 \exp(-0.4jx_4) - x_1 \exp(-0.2jx_2) - x_3 \exp(-0.4jx_4)]^2,$$

$$x^0 = (0.5 \ 0 \ 2.5 \ 3)^T, x^* = (1 \ 1 \ 2 \ 2)^T.$$

Функция невыпуклая, с искривленным оврагом, обусловленность велика.

Задача 13 (функция Вуда):

$$F(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1),$$

$$x^0 = (-3 \ -1 \ -3 \ -1)^T,$$

$$x^* = (1 \ 1 \ 1 \ 1)^T, F(x^*) = 0.$$

Функция имеет несколько локальных минимумов, это обстоятельство может вызвать преждевременное окончание процесса.

Задача 14 (функция Степенная):

$$F(x) = (10(x_1 - x_2)^2 + (x_1 - 1)^2)^4,$$

$$x^0 = (-1.2 \ 0)^T, x^* = (1 \ 1)^T, F(x^*) = 0.$$

В таблице 1 приводятся результаты численных исследований, выполненных в трех версиях предложенного ньютоновского метода безусловной минимизации: Nmbm, NmbmApp [6, 9, 14, 27, 28] и предлагаемой NmbmsApp.

Таблица 1

Результаты численных исследований задач 12, 13, 14

Table 1

The results of numerical studies for problems 12, 13, 14

Метод	Функция	k	δF	δx	k ₀
NmbmApp	Поляка	40	$4.23 \cdot 10^{-33}$	0	1341
NmbmsApp	Поляка	26	$4.23 \cdot 10^{-33}$	0	837
Nmbm	Вуда	13	0	0	36
NmbmApp	Вуда	14	$4.4 \cdot 10^{-27}$	$2 \cdot 10^{-14}$	470
NmbmsApp	Вуда	16	0	0	325
Nmbm	Степенная	12	0	0	163
NmbmApp	Степенная	13	$4.3 \cdot 10^{-61}$	$8 \cdot 10^{-9}$	323
NmbmsApp	Степенная	13	$4.81 \cdot 10^{-61}$	$8 \cdot 10^{-9}$	232

Результаты данного сравнительного исследования позволяют оценить предлагаемую версию ньютоновского метода безусловной оптимизации с регулировкой шага, основанной на факторизации Холецкого с конечно-разностной аппроксимацией

первых и вторых производных (вычислительная схема 3 модифицированного разложения Холесского) как более предпочтительную по сравнению с NmbmApp (вычислительная схема 2 модифицированного разложения Холесского).

Долан и Морэ впервые предложили применение бенчмаркинга в качестве нового инструмента использования статистических данных для сравнения итерационных алгоритмов путем демонстрации профилей производительности [30]. Термин «бенчмаркинг» англоязычный и не имеет дословного перевода на русский язык, происходит от английских слов «bench» (уровень, высота) и «mark» (отметка). Это словосочетание трактуется по-разному: «опорная отметка», «отметка высоты», «эталонное сравнение» и т.д. Как правило, термином «бенчмаркинг» обозначается один из инструментов совершенствования деятельности.

Профилем производительности P_s для метода решения оптимизационной задачи называется функция распределения какого-либо измеримого показателя производительности [30]. Вычисление профилей производительности позволяет визуализировать различия по эффективности нескольких оптимизационных методов, причем для их построения используется отдельная программа [30].

Функция $P_s(\tau)$ определяется следующим образом: $P_s(\tau) = \frac{1}{n_p} \text{size} \{ p \in P : r_{p,s} \leq \tau \}$,

$$r_{p,s} = \frac{t_{p,s}}{\min \{ t_{p,s} : s \in S \}}.$$

Здесь S – множество сравниваемых методов, P – множество решаемых с помощью этих методов задач. Количество элементов в P обозначено через n_p , соответственно n_s – количество элементов в S . В качестве измеримых показателей производительности $t_{p,s}$, как правило, используются число итераций и число вычислений значений функции (или число вычислений значений функции и оценок градиента) [31–33].

В этой технике можно выбрать индекс производительности в качестве меры сравнения среди рассмотренных алгоритмов и проиллюстрировать полученные результаты с помощью профилей производительности.

Бенчмаркинг, как и любой другой инструмент для сравнения итерационных алгоритмов, имеет свои достоинства и недостатки. Чувствительность только к измеримому показателю производительности $t_{p,s}$ и отсутствие удовлетворительного способа сравнения между собой нескольких методов оптимизации по результатам численных экспериментов [25] часто являются причинами разногласий при интерпретации результатов [30]. Кроме того, вычисление профилей производительности требует отбрасывания задач, для которых машинные реализации соответствующих методов потер-

пели неудачу [30], что усложняет процесс сравнения итерационных алгоритмов.

В данной работе, как и в [31–33], все тестовые задачи описаны в [20], все сравниваемые алгоритмы безусловной оптимизации сходятся к одному и тому же локальному решению, сравнению подлежат подходы к заданию направления спуска, результаты решения заносятся в таблицы.

Следует отметить: если объем статистических данных небольшой, то для сравнения итерационных алгоритмов, как правило, достаточно таблиц [32, 33]; иначе для сравнения итерационных алгоритмов используется бенчмаркинг путем демонстрации профилей производительности [30, 31].

В [31–33] завершение итераций производится с использованием одного из следующих критериев останова:

$$\|h^k\| \leq 10^{-5}, \|h^k\| \leq 10^{-6}, \|h^k\| \leq 10^{-6} \|h^0\| \quad (13)$$

соответственно. Отсюда следует отсутствие статистических данных о точности решения тестовых задач, что является недостатком сравнения итерационных алгоритмов. Поэтому в данной работе предлагается приводить и статистические данные о решении с максимальной точностью, а не только данные, соответствующие критериям (13). Результаты численных исследований приведены в таблицах 2 и 3 соответственно.

Таблица 2

Результаты численных исследований,
соответствующие критериям (13)

Table 2

The results of numerical studies that meet the criteria (13)

Функция	n	k	k_1
Extended Rosenbrock function	2	24	84
Extended Rosenbrock function	4	24	82
Extended Rosenbrock function	6	24	72
Wood function	4	16	28
Generalized Rosenbrock function	2	24	84
Generalized Rosenbrock function	3	23	56
Generalized Rosenbrock function	4	25	57
Extended White & Holst function	2	24	26
Extended White & Holst function	4	25	64
Extended White & Holst function	6	26	106
Extended Penalty function	2	10	101
Extended Penalty function	3	14	90
Extended Penalty function	4	13	83
Perturbed Quadratic function	2	5	24
Perturbed Quadratic function	3	5	28
Perturbed Quadratic function	4	5	0
Raydan 1 function	2	5	0
Raydan 1 function	3	5	0
Raydan 1 function	4	5	0
Raydan 2 function	2	5	0
Raydan 2 function	3	5	0
Raydan 2 function	4	5	0
Diagonal 1 function	2	5	22
Diagonal 1 function	3	6	24
Diagonal 1 function	4	6	24
Diagonal 2 function	2	6	22
Diagonal 2 function	3	6	22
Diagonal 2 function	4	7	22
Diagonal 3 function	2	7	22
Diagonal 3 function	3	7	22
Diagonal 3 function	4	7	22

Таблица 3

**Результаты численных исследований
с максимальной точностью**

Table 3

The results of numerical studies with maximum precision

Функция	n	k	k_1
Extended Rosenbrock function	2	18	18
Extended Rosenbrock function	4	19	16
Extended Rosenbrock function	6	18	26
Wood function	4	12	8
Generalized Rosenbrock function	2	18	18
Generalized Rosenbrock function	3	20	18
Generalized Rosenbrock function	4	20	15
Extended White & Holst function	2	21	26
Extended White & Holst function	4	22	20
Extended White & Holst function	6	20	33
Extended Penalty function	2	7	40
Extended Penalty function	3	10	23
Extended Penalty function	4	11	61
Perturbed Quadratic function	2	2	0
Perturbed Quadratic function	3	2	0
Perturbed Quadratic function	4	4	0
Raydan 1 function	2	3	0
Raydan 1 function	3	3	0
Raydan 1 function	4	4	0
Raydan 2 function	2	4	0
Raydan 2 function	3	4	0
Raydan 2 function	4	4	0
Diagonal 1 function	2	3	0
Diagonal 1 function	3	3	1
Diagonal 1 function	4	4	2
Diagonal 2 function	2	4	0
Diagonal 2 function	3	4	0
Diagonal 2 function	4	4	0
Diagonal 3 function	2	4	0
Diagonal 3 function	3	4	0
Diagonal 3 function	4	4	0

Заключение

В работе доказано, что интеграция техники исключения Гаусса и факторизации Холецкого определяет решение проблемы масштабирования шагов при спуске, а следовательно, и аппроксимацию неквадратичными функциями, и интеграцию с методом доверительной окрестности [1], а также то, что интеграция техники исключения Гаусса и факторизации Холецкого определяет и подходы к дальнейшему уменьшению значения нормы поправки E^k , если использовать симметричные перестановки строк и столбцов H^k следующим образом: на очередном, i -м, шаге факторизации в качестве i -й строки и i -го столбца надо брать ту из нетронутых пар, для которой величина $|c_{ii}^k| + |c_i^k|$ максимальна [6].

Кроме того, доказано, что модифицированная факторизация Холецкого – оптимизированный алгоритм в том смысле, что параметр β подбирается в нем путем минимизации априорной оценки нормы поправки E^k при условии сохранения существенно положительно определенной матрицы неизменной и с очевидностью зависит от способа задания направления спуска [1, 6]. В данной работе доказанные утверждения позволили модифициро-

вать формулу вычисления величины β , предложенную в [2], и построить вычислительную схему 3. Полученные результаты хорошо согласуются с численными экспериментами, приведенными в данной работе.

Изучен подход к увеличению эффективности ньютоновских методов с конечно-разностной аппроксимацией первых и вторых производных. Подход является основой для дальнейших исследований, результаты которых могут быть использованы для построения численных методов ньютоновского типа.

Рассмотрена взаимосвязь подхода к увеличению эффективности гауссова исключения для разреженных матриц [26] и предлагаемого подхода к увеличению эффективности численных методов ньютоновского типа – использование структуры матрицы, то есть информации о том, в каких позициях матрицы хранятся ненулевые элементы. Для ньютоновских методов безусловной оптимизации, основанных на факторизации Холецкого, с регуляризацией шага и с конечно-разностной аппроксимацией первых и вторых производных, это возможность уменьшения числа вычислений функции путем формирования матрицы вторых производных в соответствии с ее структурой. Данное исследование является прямым продолжением работы [14].

Литература

1. Зеленков Г.А., Хакимова А.Б. Подход к разработке алгоритмов ньютоновских методов оптимизации, программная реализация и сравнение эффективности // Компьютерные исследования и моделирование. 2013. Т. 5. № 3. С. 367–377.
2. Gill P.E. and Murray W. Newton-type methods for unconstrained and linearly constrained optimization. Math. Prog. 1974, no. 28, pp. 311–350.
3. Гилл Ф., Мюррей У., Райт М. Практическая оптимизация; [пер. с англ.]. М.: Мир, 1985. 509 с.
4. Гилл Ф., Мюррей У. Численные методы условной оптимизации; [пер. с англ.]. М.: Мир, 1977. 290 с.
5. Парлетт Б. Симметричная проблема собственных значений. Численные методы; [пер. с англ.]. М.: Мир, 1983. 382 с.
6. Свириденко А.Б. Априорная поправка в ньютоновских методах оптимизации // Компьютерные исследования и моделирование. 2015. Т. 7. № 4. С. 835–863.
7. Хакимова А.Б., Зеленков Г.А. Увеличение эффективности ньютоновских методов минимизации. Вычисление длины шага // Динамика неоднородных систем: тр. ИСА РАН. 2010. Вып. 14-А. Т. 53. С. 115–120.
8. Хакимова А.Б., Дикусар В.В., Зеленков Г.А. Увеличение эффективности ньютоновских методов оптимизации. Информационно-динамический подход // Динамика неоднородных систем: тр. ИСА РАН. 2010. Вып. 14-А. Т. 53. С. 97–114.
9. Свириденко А.Б., Зеленков Г.А. Управление процессом построения ньютоновских методов безусловной оптимизации: монография. Новороссийск: Изд-во ГМУ им. адм. Ф.Ф. Ушакова, 2016. 98 с.
10. Пшеничный Б.Н., Данилин Ю.М. Численные методы в экстремальных задачах. М.: Наука, 1975. 320 с.
11. Полак Э. Численные методы оптимизации. Единый подход. М.: Мир, 1974. 376 с.
12. Карманов В.Г. Математическое программирование. М.: Наука, 1975. 272 с.
13. Stepleman R.S., Winarsky N.D. Adaptive numerical differentiation. Mathematics of Computation, 1979, no. 33, pp. 1257–1264.
14. Свириденко А.Б., Зеленков Г.А. Взаимосвязь и реализа-

ция квазиньютоновских и ньютоновских методов безусловной оптимизации // Компьютерные исследования и моделирование. 2016. Т. 8. №. 1. С. 55–78.

15. Averick B.M., Carter R.G., Moré J.J. The MINPACK-2 test problem collection (Preliminary version). Math. and Comp. Sc. Divis., 1991, no. 150.

16. Averick B.M., Carter R.G., Moré J.J., Xue G.L. The MINPACK-2 test problem collection. Math. and Comp. Sc. Divis., 1992.

17. Jackson R.H.F., and Mulvey J.M. A Critical review of comparisons of mathematical programming algorithms and software (1953–1977). J. Research of the National Bureau of Standards, 1978, no. 83, pp. 563–584.

18. Gillsinn J., Hoffman K., Jackson R.H.F., Leyendecker E., Saunders P., and Shier D. Methodology and analysis for comparing discrete linear L1 approximation codes. Communications in Statistics, 1977, no. 136, pp. 399–413.

19. Crowder H.P., Dembo R.S., and Mulvey J.M. On reporting computational experiments with mathematical software. ACM Transactions on Mathematical Software, 1980, no. 5, pp. 193–203.

20. Andrei N. An unconstrained optimization test functions collection. Adv. Model. Optim. 2008, vol. 1, no. 10, pp. 147–161.

21. Bongartz I., Conn A.R., Gould N.I.M., Toint P.L., CUTE: constrained and unconstrained testing environments. ACM TOMS, 1995, vol. 21, pp. 123–160.

22. Moré J.J., Garbow B.S., Hillstom K.E. Testing unconstrained optimization software. ACM Trans. Math. Soft., 1981, vol. 7, pp. 17–41.

23. Himmelblau D., Applied nonlinear programming. McGraw-Hill, NY, 1972.

24. Lee D. A fast and robust unconstrained optimization method requiring minimum storage. Mathematical Programming, 1985, vol. 32, pp. 41–68.

25. Поляк Б.Т. Введение в оптимизацию. М.: Наука, 1983. 384 с.

26. Соловьев С.А. Решение разреженных систем линейных уравнений методом Гаусса с использованием техники аппроксимации матрицами малого ранга // Вычислительные методы и программирование. 2014. Т. 15. С. 441–460.

27. Свириденко А.Б. MNB (Ньютоновский метод безусловной оптимизации). Свид. о гос. регистр. прогр. для ЭВМ № 2015610399. 2015.

28. Свириденко А.Б., Зеленков Г.А. MNBApp (Ньютоновский метод безусловной оптимизации с численным вычислением первых и вторых производных). Свид. о гос. регистр. прогр. для ЭВМ № 2015610347. 2015.

29. Черноуцкий И.Г. Методы оптимизации. Компьютерные технологии. СПб: БХВ-Петербург, 2011. 384 с.

30. Dolan E., Moré J.J. Benchmarking optimization software with performance profiles. Math. Program., 2002, no. 91, pp. 201–213.

31. Ahookhosh M., Amini K. An efficient nonmonotone trust-region method for unconstrained optimization. Numer. Algor., 2012, no. 59, pp. 523–540.

32. Runak M., Abdula, Abbas Y. Al Bayati. A new hybrid scaled search direction for unconstrained optimization. Jour. of Kirkuk University, 2007, vol. 2, no. 1, pp. 62–81.

33. Abbo K.K. New CG Method for large-scale unconstrained optimization based on Nazareth theorem. Iraqi Jour. of Statistical Sciences, 2008, vol. 13, pp. 53–65.

Software & Systems

DOI: 10.15827/0236-235X.118.270-281

Received 10.10.16

2017, vol. 30, no. 2, pp. 270–281

REALIZATION AND TESTING OF NEWTON METHODS FOR UNCONSTRAINED OPTIMIZATION

A.B. Sviridenko¹, Lecturer, roshechka@gmail.com

¹ Novorossiysk branch of the Kuban State University, Geroev Desantnikov St. 87, Novorossiysk, 353900, Russian Federation

Abstract. The paper describes an approach to increasing the effectiveness of Newton's methods for unconstrained optimization based on the Cholesky factorization with step adjustment and finite-difference approximation of the first and second derivatives. The modified Cholesky decomposition of the second derivative matrix is the basis of increasing the efficiency of the Newton's methods. It defines the solution to the problem of scaling steps during lowering; the approximation by non-quadratic functions; integration with the method of confidence neighborhoods; decrease in norms of priori amendment. The work investigates the possibility of reducing the number of function evaluations by forming the second derivative matrix in accordance with its structure.

The article examines the relationship of the approach to increasing the efficiency of the Gaussian exceptions for sparse matrices and the proposed approach to increasing the efficiency of numerical methods of Newton type (using a matrix structure), that is information about matrix positions which store non-zero elements. For Newton's methods for unconstrained optimization based on the Cholesky factorization with step adjustment and the finite-difference approximation of the first and second derivatives it is the possibility of reducing the number of function evaluations by forming a matrix of the second derivatives in accordance with its structure.

The paper contains the descriptions of program implementations, all versions of the algorithms are implemented in Visual Basic .NET, the development environment is Microsoft Visual Studio 2010. The article shows the results of numerical studies of the effectiveness of the developed algorithms given the set of rules described in the work.

The work studies an approach to increasing the efficiency of Newton's methods with finite-difference approximation of the first and second derivatives. This is the basis for further research, the results of which might be used to build Newton-type numerical methods.

Keywords: newton's methods, matrix structure, finite-difference approximation, Cholesky's decomposition, scaling steps, method of confidence neighborhoods, norm of the prior correction, benchmarking, performance profile, collection of test problems.

References

1. Zelenkov G.A., Khakimova A.B. Approach to development of algorithms of Newtonian methods of unconstrained optimization, their software implementation and benchmarking. *Kompyuternye issledovaniya i modelirovanie* [Computer Research and Modeling]. 2013, vol. 5, no. 3, pp. 367–377 (in Russ.).

2. Gill P.E., Murray W. Newton-type methods for unconstrained and linearly constrained optimization. *Math. Prog.* 1974, no. 28, pp. 311–350.
3. Gill P., Murray U., Wright M. *Practical Optimization*. Academic Press, 1981 (Russ. ed.: Moscow, Mir Publ., 1985, 509 p.).
4. Gill P.E., Murray W. *Numerical methods of constrained optimization*. Academic Press, 1974 (Russ. ed.: Moscow, Mir Publ., 1977, 290 p.).
5. Parlett B. *The Symmetric Eigenvalue Problem*. Numerical methods. Univ. of California, 1980 (Russ. ed.: Moscow, Mir Publ., 1983, 384 p.).
6. Sviridenko A.B. The correction to Newton's methods of optimization. *Kompyuternye issledovaniya i modelirovanie* [Computer Research and Modeling]. 2015, vol. 7, no. 4, pp. 835–863 (in Russ.).
7. Khakimova A.B., Zelenkov G.A. Increasing the efficiency of Newton's minimization techniques. Calculating a step length. *Trudy ISA RAN "Dinamika neodnorodnykh sistem"* [Proc. ISA RAS "Dynamics of Inhomogeneous Systems"]. Moscow, Librokom Publ., 2010, vol. 53, pp. 115–120 (in Russ.).
8. Khakimova A.B., Dikumar V.V., Zelenkov G.A. Increasing the efficiency of Newton minimization techniques. Informodynamic approach. *Trudy ISA RAN "Dinamika neodnorodnykh sistem"* [Proc. of ISA RAS "Dynamics of Inhomogeneous Systems"]. Moscow, Librokom Publ., 2010, vol. 53, pp. 97–114 (in Russ.).
9. Sviridenko A.B., Zelenkov G.A. *Upravlenie protsessom postroyeniya nyutonovskikh metodov bezuslovnoy optimizatsii* [Managing the Process of Constructing Newton's Methods for Unconstrained Optimization]. Monograph. Novorossiysk, Admiral Ushakov Maritime State Univ. Publ., 2016, 98 p.
10. Pshenichny B.N., Danilin Yu.M. *Chislennyye metody v ekstremalnykh zadachakh* [Numerical methods in Extreme Problems]. Moscow, Nauka Publ., 1975.
11. Polak E. *Computational Methods in Optimization: A Unified Approach*. Academic Press, 1971, 329 p. (Russ. ed.: Ershko F.I. Moscow, Mir Publ., 1974, 376 p.).
12. Karmanov V.G. *Matematicheskoe programmirovaniye* [Mathematical Programming]. Moscow, Mir Publ., 1975.
13. Stepleman R.S., Winarsky N.D. Adaptive numerical differentiation. *Mathematics of Computation*. 1979, no. 33, pp. 1257–1264.
14. Sviridenko A.B., Zelenkov G.A. Correlation and realization of quasi-Newton methods of absolute optimization. *Kompyuternye issledovaniya i modelirovanie* [Computer Research and Modeling]. 2016, vol. 8, no. 1, pp. 55–78 (in Russ.).
15. Averick B.M., Carter R.G., Moré J.J. *The MINPACK-2 Test Problem Collection* (Preliminary version). Math. and Comp. Sc. Divis. 1991, no. 150.
16. Averick B.M., Carter R.G., Moré J.J., Xue G.L. The MINPACK-2 test problem collection. Math. and Comp. Sc. Divis., 1992.
17. Jackson R.H.F., Mulvey J.M. A Critical Review of Comparisons of Mathematical Programming Algorithms and Software (1953–1977). *J. Research of the National Bureau of Standards*. 1978, no. 83, pp. 563–584.
18. Gillsinn J., Hoffman K., Jackson R.H.F., Leyendecker E., Saunders P., Shier D. Methodology and Analysis for Comparing Discrete Linear L1 Approximation Codes. *Communications in Statistics*. 1977, no. 136, pp. 399–413.
19. Crowder H.P., Dembo R.S., Mulvey J.M. On Reporting Computational Experiments with Mathematical Software. *ACM Trans. on Mathematical Software*. 1980, no. 5, pp. 193–203.
20. Andrei N. An unconstrained optimization test functions collection. *Adv. Model. Optim.* 2008, no. 10 (1), pp. 147–161.
21. Bongartz I., Conn A.R., Gould N.I.M., Toint P.L. *CUTE: Constrained and Unconstrained Testing Environments*. ACM TOMS Publ., 1995, vol. 21, pp. 123–160.
22. Moré J.J., Garbow B.S., Hillstom K.E. Testing unconstrained optimization software. *ACM Trans. Math. Soft.* 1981, vol. 7, pp. 17–41.
23. Himmelblau D. *Applied Nonlinear Programming*. McGraw-Hill, NY, 1972.
24. Lee D. A fast and robust unconstrained optimization method requiring minimum storage. *Mathematical Programming*. 1985, vol. 32, pp. 41–68.
25. Polyak B.T. *Vvedenie v optimizatsiyu* [Introduction to Optimization]. Moscow, Nauka Publ., 1983.
26. Solov'ev S.A. Solution of sparse systems of linear equations using the Gauss method and low-rank approximations matrices. *Vychislitelnye metody i programmirovaniye* [Numerical Methods and Programming]. Moscow, 2014, vol. 15, pp. 441–460 (in Russ.).
27. Sviridenko A.B. *MNB (Nyutonovskiy metod bezuslovnoy optimizatsii)* [MNB (Newton's Method for Unconstrained Optimization)]. State registration Certificate of the computer program № 2015610399. 2015 (in Russ.).
28. Sviridenko A.B., Zelenkov G.A. *MNBApp (Nyutonovskiy metod bezuslovnoy optimizatsii s chislennym vychisleniem pervykh i vtorykh proizvodnykh)* [MNBApp (Newton's Method for Unconstrained Optimization with Numerical Calculation of the First and Second Derivatives)]. State registration Certificate of the computer program № 2015610347. 2015 (in Russ.).
29. Chernorutsky I.G. *Metody optimizatsii. Kompyuternyye tekhnologii* [Optimization Methods. Computer Technologies]. St. Petersburg, BHV-Peterburg Publ., 2011, 384 p.
30. Dolan E., Moré J.J. Benchmarking optimization software with performance profiles. *Math. Program.* 2002, no. 91, pp. 201–213.
31. Ahookhosh M., Amini K. An efficient nonmonotone trust-region method for unconstrained optimization. *Numer. Algor.* 2012, no. 59, pp. 523–540.
32. Abdula R.M., Al Bayati A.Y. A new hybrid scaled search direction for unconstrained optimization. *Jour. of Kirkuk Univ.* 2007, vol. 2, no. 1, pp. 62–81.
33. Abbo K.K. New CG Method for Large-Scale Unconstrained Optimization Based on Nazareth theorem. *Iraqi Jour. of Statistical Sciences*. 2008, vol. 13, pp. 53–65.