где i_0 – номер столбца под $\bar{x}_{i_0}$ ($i_0 \in \bar{I}_0$) и величину $Q(x)$

$$Q'(x) = Q(x) + c_{i_0} \cdot d_{i_0}.$$

После этого столбцы, соответствующие $\bar{x}_i$ ($i \in \bar{I}_0$), исключаются и решается задача линейного программирования с данными, указанными в полученной таблице, согласно вышеизложенному алгоритму К-симплексным методом. Решение данной задачи может быть представлено в виде

$$x^* = (\bar{x}_1^* + A_1, \bar{x}_2^* + A_2, \ldots, \bar{x}_n^* + A_n);$$

$$Q^* = \sum_{i=1}^{n} c_i x_i^*.$$

Б.Ю.Лемешко, Е.Б.Цой

О СКОРОСТИ СХОДИМОСТИ МЕТОДА СОПРЯЖЕННОГО ГРАДИЕНТА

В работе [I] доказывается, что метод сопряженного градиента Флетчера-Ривса, так же, как и метод сопряженного градиента Полака – Рибера, имеет квадратичную через n шагов скорость сходимости даже при неточном одномерном поиске.

Цель данной работы – экспериментальное исследование скорости сходимости метода Флетчера-Ривса в зависимости от точности одномерного поиска.

Последовательность операций этого алгоритма описывается следующим образом. Пусть $f: E_n \to E_1$ – непрерывная, дважды дифференцируемая функция. Выбираем произвольную точку $\bar{x}_0 \in E_n$. Обозначим через P_k направление одномерного поиска на k-м шаге. Полагаем $P_0 = \nabla f(\bar{x}_0)$ (I)

Для $k = 0, 1, 2, \ldots$ определяем

$$\bar{x}_{k+1} = \bar{x}_k - \alpha_k P_k , \tag{2}$$

$$P_{k+1} = \nabla f(\bar{x}_{k+1}) + \beta_k P_k , \tag{3}$$

где α_k – наименьший положительный корень уравнения

$$(\nabla f(\bar{x}_k - \alpha P_k), P_k) = 0 , \tag{4}$$

а

$$\beta_k = \begin{cases} 0, \text{ если } k = n, 2n, 3n, \ldots \\ \|\nabla f(\bar{x}_{k+1})\|^2 / \nabla f(\bar{x}_k)\|^2 \end{cases} \tag{5}$$

–в остальных случаях.

Из (5) видно, что процедура возобновляется после каждых n итераций.

Величина α_k в данном случае определялась в результате одномерного поиска по направлению P_k, который выполнялся одношаговой и многошаговой квадратичной интерполяцией и методом "золотого сечения". При использовании одношаговой квадратичной интерполяции выделялось такое значение α, при котором выполнялись следующие неравенства

$$f(\bar{x}_k) \geq f(\bar{x}_k - \alpha P_k) ,$$

$$f(\bar{x}_k - 2\alpha P_k) \geq f(\bar{x}_k - \alpha P_k), \tag{6}$$

причем хотя бы одно из них должно выполняться строго. Тогда α_k находится по формуле

$$\alpha_k = \frac{\alpha}{2} \cdot \frac{3f(\bar{x}_k) - 4f(\bar{x}_k - \alpha P_k) + f(\bar{x}_k - 2\alpha P_k)}{f(\bar{x}_k) - 2f(\bar{x}_k - \alpha P_k) + f(\bar{x}_k - 2\alpha P_k)} . \tag{7}$$

В случае многошаговой квадратичной интерполяции α_k уточняется по аналогичной формуле. В программе, реализованной на языке АЛГОЛ для ЭВМ "Одра-1204", $\nabla f(\bar{x}_k)$ оценивался через конечные разности, причем шаг дифференцирования регулировался автоматически. Исследования проводились на функциях:

$$f_1(\bar{x}) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2,$$

$$f_2(\bar{x}) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + 90(x_3 - x_4)^2 + 10.1\left((x_2 - 1)^2 + (x_4 - 1)^2\right) + 19.8(x_2 - 1)(x_4 - 1)$$

$$f_3(\bar{x}) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4,$$

$$f_4(\bar{x}) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^4 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4,$$

$$f_5(\bar{x}) = (e^{x_1} - x_3)^4 + 100(x_2 - x_3)^6 + tg^4(x_3 - x_4) + x_1^8 + (x_4 - 1)^2.$$

Скорость сходимости метода оценивалась на основании теоремы о квадратичной через n шагов скорости сходимости [I] и сравнения с модифицированным методом Ньютона([3] , стр.199).

Некоторые результаты вычислений приведены в таблице.

Метод	f_1			f_2			f_3			f_4			f_5		
	$\bar{x}_0 = (-1,21)'$			$\bar{x}_0 = (-3,-1,-3,-1)'$			$\bar{x}_0 = (3,-1,0,1)'$			$\bar{x}_0 = (3,-1,0,1)'$			$\bar{x}_0 = (6,5,4,3)$		
	K	N_f	f_1^*	K	N_f	f_2^*	K	N_f	f_3^*	K	N_f	f_4^*	K	N_f	f_5^*
I	16	220	$2.8\cdot10^{-10}$	34	519	$1.7\cdot10^{-7}$	36	709	$5.2\cdot10^{-6}$	18	304	$1.7\cdot10^{-5}$	25	425	$7.9\cdot10^{-7}$
II	16	302	$2.7\cdot10^{-7}$	34	729	$1.7\cdot10^{-7}$	40	755	$2.4\cdot10^{-6}$	60	1138	$1.2\cdot10^{-5}$	25	564	$6.07\cdot10^{-6}$
III	13	357	$4.4\cdot10^{-11}$	$\frac{46}{33}$	$\frac{1144}{944}$	$\frac{3.2\cdot10^{-7}}{1.1\cdot10^{-9}}$	37	1090	$5.7\cdot10^{-6}$	60	1751	$1.3\cdot10^{-5}$	30	902	$9.\cdot10^{-6}$
IV	35	300	$6\cdot10^{-9}$												
V				$\frac{22}{23}$	–	$\frac{1.1\cdot10^{-6}}{1.12\cdot10^{-12}}$									

В таблице обозначено: K – число итераций, N_f – общее количество вычислений функции, f_i^* – достигнутое значение функции.

Здесь I – алгоритм с одношаговой квадратичной интерполяцией при одномерном поиске;

II – алгоритм с многошаговой квадратичной интерполяцией;

III – алгоритм с методом "золотого сечения";

IУ – метод Розенброка [2] ;

У – модифицированный метод Ньютона [3] после 22 и 23 итераций.

В строке III функции f_2 знаменатель соответствует более точному одномерному поиску.

Анализ вычислений показал хорошее согласие с результатами работы [I] . Влияние точности одномерного поиска на скорость сходимости всего алгоритма особенно сильно проявилось при использовании метода "золотого сечения". Увеличение точности в этом случае приводило не только к сокращению числа итераций, но и к общему уменьшению объема вычислений. Наиболее предпочтительным оказалось использование одношаговой квадратичной интерполяции, которая характеризовалась наименьшими вычислительными затратами. При этом было отмечено, что эффективность алгоритма мала, если функция имеет плоский экстремум.

Литература

I. Kawamura Kazuhiko, Volz Richard A. On the rate of convergence of the conjugate gradient reset method with inaccurate linear minizations. "IEEE Trans. Automat. Contr." 1973, 18, №4 pp. 360-366

2. Kovács Zsoltne, Zs. Kovács. "Optimalizálási eljárások" "Hiradástechnika", 1970, 21, №11 pp. 322-335.

3. Фиакко А., Мак-Кормик Г. Нелинейное программирование. Методы последовательной безусловной минимизации. М., "Мир", 1972.