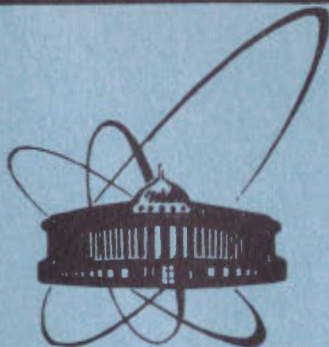


18/6/84



ОБЪЕДИНЕННЫЙ
ИНСТИТУТ
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА

P11-84-188

О.Е.Горчаков

ПРЕИМУЩЕСТВА МЕТОДА
ОРТОГОНАЛИЗАЦИИ ГРАММА-ШМИДТА
ПО СРАВНЕНИЮ С МЕТОДОМ ФОРСАЙТА
ПРИ МНОГОМЕРНОЙ АППРОКСИМАЦИИ

Направлено в "Журнал вычислительной математики
и математической физики"

1984

Как известно, при аппроксимации экспериментальных данных довольно часто применяется метод ортогональных полиномов. Используются две разновидности этого метода. Первый - метод ортогонализации Грамма-Шмидта /1/, который позволяет из некоторой системы полиномов $\{f_k\}$ построить систему полиномов $\{\psi_k\}$, ортогональных на данном множестве

$$\psi_k = f_k - \sum_{j=1}^{k-1} \frac{(f_k, \psi_j)}{(\psi_j, \psi_j)} \psi_j. \quad /1/$$

Этот метод для целей аппроксимации экспериментальных данных был использован в /2-5/. Второй метод - использование рекуррентных соотношений Форсайта /6/ /одномерный вариант/ и Вайсфелда /7/ /многомерный вариант/ в случае, когда $\{f_k\}$ - алгебраические одночлены. При этом в рекуррентном соотношении для ψ_k присутствует число функций ψ меньше, чем $k-1$. Эти соотношения были использованы в /8-15/. Из анализа литературы следует, что метод Форсайта /Вайсфелда/ применяется значительно чаще, к тому же в /3,4/ метод /1/ реализован на языке символьного программирования REDUCE-2 /16/, и ортогонализация проводится на квадрате $\langle -1, 1 \rangle \times \langle -1, 1 \rangle$. В одной из этих работ /4/ утверждается, что метод Грамма-Шмидта в общем случае слишком громоздок. В /17/ есть утверждение, что метод Форсайта значительно проще по сравнению с методом Грамма-Шмидта. В /2/ эти два метода не сравниваются, а выбор метода Грамма-Шмидта обусловлен тем, что в качестве системы полиномов $\{f_k\}$ используются как алгебраические одночлены, так и полиномы Чебышева и Лежандра. Поэтому, казалось бы, можно сделать вывод о том, что в случае, когда $\{f_k\}$ - алгебраические одночлены, метод Форсайта /Вайсфелда/ проще и эффективнее по сравнению с методом Грамма-Шмидта. В настоящей работе показывается обратное - метод Грамма-Шмидта в применении к аппроксимации экспериментальных данных является более простым и требует значительно меньшего числа операций, чем метод Форсайта /Вайсфелда/.

Пусть для аппроксимации экспериментальных данных $\{y_i\}$ и $\{x_{1i}, x_{2i}, \dots, x_{ni}\}$, где $i=1 \div N$, мы выбрали некоторую систему полиномов $\{f_k\}$, $k=1 \div M$.

Тогда методом наименьших квадратов из минимума функционала

$$S = (\vec{y} - \sum_{j=1}^M c_j \vec{f}_j)^2 \quad /2/$$

где $\vec{f}_j = \{f_j(\vec{x}_1), f_j(\vec{x}_2), \dots, f_j(\vec{x}_N)\}$, можно найти оценку параметров $\{c_j\}$. Используя метод Грамма-Шмидта, определим функции $\{\psi_j\}$ следующим образом:

$$\vec{\psi}_1 = \vec{f}_1, \quad \vec{\psi}_j = \vec{f}_j - \sum_{k=1}^{j-1} \frac{(\vec{f}_j, \vec{\psi}_k)}{(\vec{\psi}_k, \vec{\psi}_k)} \vec{\psi}_k. \quad /3/$$

Из такой записи следует, что $(\vec{\psi}_i, \vec{\psi}_j) = 0$ при $i \neq j$. Тогда функционал /2/ можно переписать в виде

$$S = (\vec{y} - \sum_{j=1}^M a_j \vec{\psi}_j)^2. \quad /4/$$

Дифференцируя по a_j и учитывая ортогональность $\{\psi_j\}$, получаем

$$a_j = \frac{(\vec{y}, \vec{\psi}_j)}{(\vec{\psi}_j, \vec{\psi}_j)}. \quad /5/$$

Этот метод реализовать программно можно, в принципе, несколькими способами. Но естественно выбрать, по возможности, самый простой и требующий наименьшего числа операций. Также всегда существует стремление к тому, чтобы программа на ЭВМ занимала минимум памяти. Если из /2,5,10/ определить максимальные значения параметров N , M и n , то они оказываются равными 10000, 60 и 10 соответственно.

Автором настоящей статьи в /5/ был использован следующий алгоритм. Для наглядности запишем выражение /3/ в несколько ином виде:

$$\begin{aligned} \vec{\psi}_1 &= \vec{f}_1 \\ \vec{\psi}_2 &= \frac{-(\vec{f}_2, \vec{\psi}_1)}{(\vec{\psi}_1, \vec{\psi}_1)} \vec{\psi}_1 + \vec{f}_2 \\ \vec{\psi}_3 &= \frac{-(\vec{f}_3, \vec{\psi}_1)}{(\vec{\psi}_1, \vec{\psi}_1)} \vec{\psi}_1 - \frac{(\vec{f}_3, \vec{\psi}_2)}{(\vec{\psi}_2, \vec{\psi}_2)} \vec{\psi}_2 + \vec{f}_3 \\ &\dots \\ &\dots \\ \vec{\psi}_M &= -\frac{(\vec{f}_M, \vec{\psi}_1)}{(\vec{\psi}_1, \vec{\psi}_1)} \vec{\psi}_1 - \frac{(\vec{f}_M, \vec{\psi}_2)}{(\vec{\psi}_2, \vec{\psi}_2)} \vec{\psi}_2 - \dots - \frac{(\vec{f}_M, \vec{\psi}_{M-1})}{(\vec{\psi}_{M-1}, \vec{\psi}_{M-1})} \vec{\psi}_{M-1} + \vec{f}_M. \end{aligned} \quad /6/$$

Отметим, что если $\{f_k\}$ - алгебраические одночлены, то существует эффективный способ вычисления всего набора значений $\{f_1(\vec{x}_i), f_2(\vec{x}_i), \dots, f_M(\vec{x}_i)\}$ в данной точке \vec{x}_i . Он основывается на том, что можно заранее определить такие ℓ , k и r , что

$$f_k(\vec{x}) = x_p f_\ell(\vec{x}), \quad /7/$$

где $\ell < k$, $1 \leq p \leq n$. При этом подразумевается, что $\{f_k\}$ упорядочены. Это можно сделать, например, следующим образом /трехмерный случай/:

$$\{f_k\} = \{1, x, y, z, x^2, xy, xz, y^2, yz, z^2, x^3, \dots\}. \quad /8/$$

Перейдем к описанию конкретного алгоритма. Вначале вычислим скалярные произведения с участием $\vec{\psi}_1$: $\vec{\psi}_1^2, (\vec{y} \vec{\psi}_1), (\vec{r}_i \vec{\psi}_1) |_{j=2 \div M}$. Для вычисления $f_j(\vec{x}_i) |_{j=2 \div M}$ используем соотношения /7/. Определим коэффициенты разложения ψ_2 по функциям $\{f_k\}$:

$$\psi_2 = \sum_{j=1}^2 b_{2j} f_j. \quad /9/$$

Затем вычислим $\vec{\psi}_2^2, (\vec{y} \vec{\psi}_2)$ и $(\vec{r}_i \vec{\psi}_2) |_{j=3 \div M}$, используя при этом соотношения /7/ и /9/. Для ψ_3 определяем $\{b_{3j}\}$ и т.д. - до функции ψ_M включительно. Из /9/ следует, что $b_{kj} = 0$ при $j > k$, и поэтому

$$\sum_{i=1}^M a_i \psi_i = \sum_{j=1}^M a_j \sum_{k=1}^j b_{jk} f_k = \sum_{k=1}^M (\sum_{j=1}^M a_j b_{jk}) f_k, \quad /10/$$

т.е.

$$c_k = \sum_{j=1}^M a_j b_{jk} = \sum_{j=k}^M a_j b_{jk}. \quad /11/$$

Для вычисления $\{c_k\}$ по данному алгоритму требуется число операций, равное $D_1 N M^2$, где $D_1 \geq 1$. Отметим, что если экспериментальные точки имеют различные веса, то это просто учесть путем введения их /весов/ в каждое скалярное произведение.

При реализации алгоритма на ЭВМ объем памяти, необходимый для хранения основного числового материала $\{y_i\}$ и $\{x_i\}$, составляет $(n+1)N$ слов /без учета массива $\{b_{ij}\}$ /.

В программе /2/ требуется еще дополнительно NM слов - для хранения величин $f_k(\vec{x}_i)$, $k=1 \div M$, $i=1 \div N$. Дополнительная память может достигать 100000-200000 слов, в то время как выигрыш за счет уменьшения числа операций, по сравнению с алгоритмом /5/, увеличится только в $\approx 1,5$ раза. Но, в свою очередь, алгоритм /5/ можно ускорить в 1,5-2 раза за счет совершенно незначительного усложнения, а именно: вначале вычислить все значения $(\vec{y} \vec{r}_i), (\vec{r}_i \vec{r}_j)$, где $i=1 \div M$ и $j=1 \div i$. Для этого требуется $D_2 N M^2$ операций, где $D_2 = D_1/2$. Для вычисления $\{c_k\}$ потребуется число операций $\sim M^3$, что обычно значительно меньше, чем $D_2 N M^2$. Такое изменение алгоритма /5/ делает величину D_2 практически независимой от вида полиномов $\{f_k\}$. Это могут быть алгебраические одночлены, полиномы Чебышева или Лежандра, и прочее. Поэтому следует признать, что алгоритм /2/ неэффективен в отношении использования памяти ЭВМ.

Теперь кратко опишем метод Форсайта /Вайсфелда/. Как уже отмечалось, этот метод относится к случаю, когда $\{f_j\}$ - алгебраические одночлены, т.е. $f_j(\vec{x}) = x_1^{j_1} x_2^{j_2} \dots x_n^{j_n}$. Определим

$$\sigma(j) = \sum_{k=1}^n j_k. \text{ Упорядочить } \{f_j\} \text{ можно следующим образом: } i < j,$$

если только $\sigma(i) \leq \sigma(j)$ и для некоторого $\ell < n$ $i_\ell + \dots + i_n < j_\ell + \dots + j_n$. Для каждого j можно определить единственное k такое, что $\sigma(j) = j_1 + \dots + j_k$ и $j_{k+1} = \dots = j_n = 0$, где $1 \leq k \leq n$.

Так как $j = j(j_1, j_2, \dots, j_n)$, то введем $K = j(j_1, j_2, \dots, j_{k-1}, j_{k+1}, \dots, j_n)$ и $L = j(\sigma(j) - 2, 0, \dots, 0)$ и тогда рекуррентное соотношение Форсайта /Вайсфелда/ запишется в виде

$$\vec{\psi}_j = c_j [(x_k - \beta_{j-1}) \vec{\psi}_K - \sum_{\substack{m=L \\ m \neq K}}^{j-1} a_j^m \vec{\psi}_m], \quad /12/$$

где

$$\beta_{j-1} = (x_k \vec{\psi}_K, \vec{\psi}_K), \quad /13/$$

$$a_j^m = (x_k \vec{\psi}_K, \vec{\psi}_m), \quad m = L \div j - 1, \quad m \neq K, \quad /14/$$

$$c_j^{-2} = (x_k \vec{\psi}_K, x_k \vec{\psi}_K) - \beta_{j-1}^2 - \sum_{\substack{m=L \\ m \neq K}}^{j-1} (a_j^m)^2. \quad /15/$$

Число функций $\{\psi_k\}$, которое присутствует в рекуррентном соотношении для ψ_j , равно примерно $j^{1-1/n}$, т.е. с увеличением n оно стремится к числу j , к тому же, что и в методе Грамма-Шмидта. Но в методе /1/ в качестве коэффициентов a_j^m выступают величины $(\vec{r}_i \vec{\psi}_m) / \vec{\psi}_m^2$, на вычисление которых затрачивается меньшее число операций. А именно - при фиксированном m вычисление всех значений $\{(\vec{r}_i \vec{\psi}_m) / \vec{\psi}_m^2\}$ требует $\sim j$ операций, а для вычисления при фиксированном j всех $\{a_j^m\}$ требуется $\sim j^{2-1/n}$ операций - это следует из необходимости вычисления $\psi_m(\vec{x}_i) |_{m=L \div j-1}$. В итоге получается, что необходимое число операций в методе Форсайта /Вайсфелда/ - $D_3 N M^{3-1/n}$ при $D_3 \approx D_1$. Для конкретной задачи с $N = 6000$, $M = 35$ и $n = 3$ алгоритм из работы /5/ требует 50 с счетного времени, а из работы /10/ - 1 ч /на ЭВМ СДС-6500/. Правда, необходимо отметить, что в /10/ метод Форсайта /Вайсфелда/ реализован не самым эффективным способом, что привело к увеличению счетного времени примерно в 2 раза. Из /12/÷/15/ видно, что этот алгоритм более громоздок; это подтверждается также тем, что число операторов фортран-IV в программах /10, 13/ в 1,5÷2 раза больше, чем в /5/.

В результате можно сделать вывод о том, что алгоритм многомерного фитирования, реализующий метод Форсайта /Вайсфелда/, является более громоздким и требует значительно большего числа операций по сравнению с методом Грамма-Шмидта.

Автор выражает благодарность В.П.Курочкину и Н.Б.Богдановой за полезные обсуждения.

ЛИТЕРАТУРА

1. Рисс Ф., Секенфальви-Надь Б. Лекции по функциональному анализу. ИИЛ, М., 1954.
2. Brun R. et al. CERN DATA HANDLING DIVISION, DD/75/23, Geneva, June, 1977.
3. Байла И., Ососков Г.А. ОИЯИ, P10-11834, Дубна, 1978.
4. Байла И., Ососков Г.А., Хэрн А.К. ОИЯИ, P10-11944, Дубна, 1978.
5. Алексеев Г.Д. и др. ОИЯИ, P1-83-894, Дубна, 1983.
6. Forsythe G.E. J.Soc.Industr.Appl.Math., 1957, vol.5, No.2, p.74-88.
7. Weisfeld M. Numerische Mathematik, 1959, vol.1, p.38-40.
8. Hudson D. LSQFIT CERN Program Library E202.
9. Pomentale T. PS11 CERN Program Library E204.
10. Pomentale T. PS12 CERN Program Library E205.
11. Гаджоков В., Богданова Н. ОИЯИ, P11-12 860, Дубна, 1979.
12. Гаджоков В., Богданова Н. ОИЯИ, P11 80-122, Дубна, 1980.
13. Гаджоков В., Богданова Н. ОИЯИ, P11-80-781, Дубна, 1980.
14. Гаджоков В. ОИЯИ, P11-80-782, Дубна, 1980.
15. Bogdanova N., Gadjokov V. Comp.Phys.Comm., 1981, vol.24, No.2, p.225-229.
16. Hern A.C. REDUCE-2 User's Manual. University of Utah, Salt Lake City, 1973.
17. Худсон Д. Статистика для физиков. "Мир", М., 1967.



Рукопись поступила в издательский отдел
26 марта 1984 года.

Горчаков О.Е. P11-84-188
Преимущества метода ортогонализации Грамма-Шмидта
по сравнению с методом Форсайта при многомерной аппроксимации

Показано, что метод ортогонализации Форсайта /Вайсфелда/, который наиболее часто используется в программах многомерного фитирования, имеет значительно меньшее быстродействие, чем метод Грамма-Шмидта.

Работа выполнена в Лаборатории ядерных проблем ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна 1984

Перевод О.С.Виноградовой

Gorchakov O.E. P11-84-188
Advantages of Gramm-Schmidt Orthogonalization Method
as Compared with the Forsythe Method at Multidimensional
Approximation

It is shown that the Forsythe (Weisfeld) method of orthogonalization which is often used in programs of multidimensional fitting, is considerably slower than that one developed by Gramm-Schmidt.

The investigation has been performed at the Laboratory of Nuclear Problems, JINR.

Preprint of the Joint Institute for Nuclear Research. Dubna 1984