ՀԱՅԱՍՏԱՆԻ ԳԻՏՈՒԹՅՈՒՆՆԵՐԻ ԱԿԱԴԵՄԻԱՅԻ ԶԵԿՈՒՅՑՆԵՐ
ДОКЛАДЫ АКАДЕМИИ НАУК АРМЕНИИ

Том 94 1993 № 1

ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА

УДК 519.64

Член-корреспондент АН Армении А. Б. Нерсесян, С. А. Цируян

Быстрый алгоритм решения многомерного уравнения с частично теплицевым ядром

(Представлено 21/V 1992)

1. Рассмотрим уравнение Фредгольма второго рода

$$(I-K)\,Y \overset{\text{def}}{=} Y(x)-\int\limits_D K(x,\,t)\,Y(t)\,dt=f(x),\quad x\in D \tag{1}$$

где $D\in R^m$ $(m\geqslant 2)$ — гиперкуб $D=\{0<x_i<1,\quad i=1,\,2,\,\ldots,\,m\}$.

$$K(x,\,t)\in C^p(\overline{D}\times\overline{D}),\quad f(x)\in C^p(\overline{D})\quad(p\geqslant 1),\quad dt=dt_1\ldots dt_m.$$

Задача решения уравнений типа (1) возникает в ряде прикладных областей (см. ([1])). По сравнению с одномерным случаем ($m=1$, $D=(a,\,b)$) ситуация здесь сложнее, прежде всего, из-за возрастающего объема вычислений и размеров требуемой оперативной памяти.

Пусть, например, выбран конкретный тип кубатурных формул p-го порядка точности (относительно наибольшего шага сети). Каждое i-ое ребро гиперкуба D дискретизируем в N_i точках $\{x_i^k\}$ ($k=1,\,2,\,\ldots,\,N_i$, $i=1,\,2,\,\ldots,\,m$). Приближенное решение $Y(x)$ будем искать на сети $x\in\{x_1^{k_1}\}\times\{x_2^{k_2}\}\times\cdots\times\{x_m^{k_m}\}$, $k_i=1,\,2,\,\ldots,\,N_i$, $i=1,\,2,\,\ldots,\,m$.

Таким образом, задача сводится к решению алгебраической системы с $(N\times N)$-матрицей, где $N=N_1N_2\ldots N_m$. Для решения этой системы (при ее однозначной разрешимости) методом Гаусса потребуется выполнить порядка $N^3/3$ мультипликативных операций с использованием оперативной памяти в N^2 ячеек ($N\gg 1$).

Понятно, что такие требования к вычислительной системе (особенно при повышенных требованиях к точности решения) не всегда выполнимы, и поэтому очень важна задача поиска средств выхода из подобных ситуаций.

В связи с этим заметим, что метод статистического моделирования (так называемый метод Монте-Карло), несмотря на свою универсальность и применимость при любом m, имеет невысокую точность и практически не улучшает своих качеств при повышении гладкости функций K и f.

2. Проблема может быть решена за счет специфических свойств оператора $\mathrm{I}-\mathrm{K}$. Здесь мы предположим, что ядро K частично теплицево (частично разностное), т. е.

$$\mathrm{K}(x, t) = \mathrm{K}(x', t', x_m - t_m), \qquad x', t' \in 0, \quad |x_m - t_m| \leqslant 1, \tag{2}$$

где $x' = (x_1, x_2, \ldots, x_{m-1}) \in D' = \{0 < x_i < 1, \quad i = 1, 2, \ldots, m-1\}$.

В классическом одномерном случае теплицева ядра $\mathrm{K} = \mathrm{K}(x-t)$ ($m=1$, $D=(0, 1)$) для решения уравнения (1) на основе квадратурных формул построены быстрые алгоритмы (см. [3,4]). Именно, если при выборе на отрезке $[0, 1]$ сети из N точек общий подход (см. выше) приводит к алгоритмам сложности $O(N^3)$ ($N \to \infty$) с требуемой памятью порядка $O(N^2)$, то соответствующие показатели быстрых устойчивых алгоритмов имеют, соответственно, порядок $O(N^2)$ и $O(N)$.

В случае двумерного ($m=2$) ядра (2) также были построены более быстрые, чем указано выше, алгоритмы (см. [4,5]), основанные на сведении задачи (1)—(2) к алгебраической системе с блочно-теплицевой матрицей.

Ниже предлагается новый подход, основанный на методах работ [4,6].

3. Пусть $D_\tau = D' \times (0, \tau)$, $0 < \tau \leqslant 1$. Обозначим через $\mathrm{I}-\mathrm{K}_\tau$ оператор (1) с $D = D_\tau$ (т. е. $\mathrm{I}-\mathrm{K} = \mathrm{I}-\mathrm{K}_1$). В случае ядер K общего вида без труда (см. [6]) можно доказать следующий результат.

Теорема 1. *Пусть операторы* $\mathrm{I}-\mathrm{K}_\tau$ *обратимы при* $\tau = \tau_1, \tau_2$ ($\tau_1 < \tau_2$). *Тогда для их резольвентных ядер* $R(x, t, \tau)$ ($(\mathrm{I}-\mathrm{K}_\tau)^{-1} = \mathrm{I} + R_\tau$) *и решений* $Y(x, \tau)$ *уравнений* $(\mathrm{I}-\mathrm{K}_\tau)Y = f$, $\tau = \tau_1, \tau_2$, $\tau_1 < \tau_2$, *справедливы соотношения*:

$$R(x, t, \tau_2) = R(x, t, \tau_1) + \int_\Delta R(x, s, \tau_1) R(s, t, \tau_2)\, ds, \tag{3}$$

$$Y(x, \tau) = Y(x, \tau) + \int_\Delta R(x, s, \tau) Y(s, \tau)\, ds, \tag{4}$$

где $\Delta = D_{\tau_1} \setminus D_{\tau_0} = D' \times (\tau_1, \tau_2)$. $\square$

Вернемся теперь к ядрам типа (2).

Теорема 2. *Пусть операторы* $\mathrm{I}-\mathrm{K}_\tau$ *с ядром* (2) *обратимы в* $C(\overline{D})$ *на сети* (τ_n) $(0 = \tau_0 < \tau \cdots < \tau_N = 1)$. *Тогда для их резольвентных ядер* $R(x, t, \tau_n) = R(x', t', x_m, t_m, n)$ *и решений* $Y(x, n)$ *уравнений* $(\mathrm{I}-\mathrm{K}_\tau)\, Y = f(x)$, $0 \leqslant x \leqslant 1$, $\lambda \in \{\tau_n\}$ *справедливы соотношения* ($h = \tau_{n+1} - \tau_n$, $n = 0, 1, \ldots, N-1$):

$$R(x', t', x_m, t_m, n+1) = R(x', t', x_m - h, t_m - h, n) +$$

$$+ \int_0^h \int_D R(x', s', x_m - h, u - h, n) R(s', t', u, t_m, n+1)\, ds' du, \quad (5)$$

$$x', t', \in \overline{D}', \quad s' = (s_1, \ldots, s_{m-1}), \quad ds' = ds_1 \ldots ds_{m-1},$$

$$\tau_{n+1} \leqslant x_m, t_m \leqslant 1, \qquad |x_m - t_m| \leqslant 1;$$

$$Y(x, n+1) = Y(x, n) + \int_{\Delta_n} R(x', s', x_m, u, n+1) Y(s', u, n)\, ds' du, \quad (6)$$

$$x = (x', x_m) \in D_1 \qquad \Delta_n = D_{n+1} \setminus D_n.$$

Доказательство. Формула (6) фактически совпадает с (4). Равенство (5) выводится из известных интегральных уравнений для резольвентных ядер с использованием вида ядра (2). Отметим лишь, что надо дополнительно обосновать справедливость формулы (5) вне гиперкуба $\overline{D}_{n+1} \times \overline{D}_{n+1}$.

4. Формулы (5) и (6) являются основой для построения рекуррентных по n параллельных алгоритмов решения уравнения (1) с ядром (2). Остановимся подробнее на случае $K, f \in C^1$, когда на равномерной сетке с одним и тем же шагом h для $\tau, x_1, x_2, \ldots, x_m$ ($h = 1/N$, $N \geqslant 2$) применяется формула прямоугольников точности $O(h)$ ($h \to 0$). Очевидно, что при $n = 0$ имеем стартовые значения:

$$R(x', t', x_m, t_m, 0) = K(x', t', x_m - t_m), \quad (5')$$

$$x', t' \in D', \quad |x_m - t_m| < 1, \quad -1 \leqslant x_m, t_m \leqslant 1.$$

$$Y(x, 0) = f(x), \qquad x \in \overline{D} = \overline{D}' \times [0, 1[. \quad (6')$$

Далее, на каждом шаге n ($n = 0, 1, \ldots, N-1$) имеем (см. (3), (5), (6)), с точностью $O(h^2)$ ($h = 1/N \to 0$, $N \to \infty$):

$$R(x', t', x_m, (n+1)h, n+1) = R(x', t', x_m - h, nh, n) +$$

$$+ h^2 \sum_{p_1 \ldots p_{m-1}=0}^{N-1} R(x', s', x_m - h, 0, n) R(s', t', h, (n+1)h, n+1) \quad (7)$$

$$x', t' \in D', \qquad (n+1)h - 1 \leqslant x_m \leqslant 1;$$

$$R(x', t', x_m, 0, n+1) = R(x', t', x_m, 0, n) +$$

$$+ h^2 \sum_{p_1 \ldots p_{m-1}=0}^{N-1} R(x', s', x_m, nh, n+1) R(s', t', h, (n+1)h, n). \quad (8)$$

$$x', t' \in D' \quad (n+1)h - 1 \leqslant x \leqslant 1;$$

$$Y(x,\ n+1 = Y(x,\ n)+$$

$$+ h^2 \sum_{p_1 \ldots p_{m-1}=0}^{N-1} R(x',\ s',\ x_m,\ (n+1)h,\ n+1)\, Y(s',\ (n+1)h,\ n), \qquad (9)$$

$$x=(x',\ x_m)\in D,$$

где $s'=(p_1 h,\ p_2 h,\ \ldots,\ p_{m-1} h)$, а x', x_m и t' меняются на равномерной сетке с шагом h.

5. Назовем алгоритмом $A(h)$ реализацию формул (7)–(9) при стартовых значениях (5') (с $t_m=0$) и (6'). Это параллельный, рекуррентный алгоритм неявного типа точности $O(h)$ ($h\to 0$, на каждом шаге точность $O(h^2)$, а всего N шагов, $h=1/N$). Неявность следует из того, что (на n-том шаге) для определения в (7) справа неизвестных значений $R(s',\ t',\ h,\ (n+1)h,\ (n+1)$ необходимо решить алгебраическую систему с матрицей размеров $Nm\times Nm$, что потребует выполнения порядка $O(N^3m^3)$ мультипликативных операций. Общая сложность алгоритма составляет $3N^{3m-1}+O(N^{3m-2})$ $(N\to\infty)$ мультипликативных операций. Память — $2N^{2m-1}-O(N)$ ячеек.

При применении на той же x-сетке алгоритма п. 1 на основе метода Гаусса с построчным выбором ведущего элемента (назовем этот подход алгоритмом $G(h)$) потребовалось бы выполнить $\frac{1}{3}N^{3m}+O(N^{3m-1})$ мультипликативных операций при объеме памяти в $N^{2m}+O(N)$ ячеек. Однако этот алгоритм последовательный, и объективно было бы провести сравнение с параллельным алгоритмом Гаусса—Жордана, имеющим сложность порядка $\frac{1}{2}N^{3m}$ при объеме памяти в N^{2m} ячеек. (Последний алгоритм обозначим $GJ(h)$).

Таким образом, при больших N сложность алгоритма $A(h)$ по сравнению с $G(h)$ уменьшается в $N/6$ раз, а объем памяти — в N раз.

6. Алгоритмы $G(h)$ и $A(h)$ были испытаны в численных экспериментах. Изучались следующие задачи ($x=(x_1,\ x_2)$, $t=(t_1,\ t_2)$).

Задача 1.

$$K=\frac{(x_2-t_2)\sin(x_1^2+t_1)}{1+x_1^2+(x_2-t_2)^2}, \qquad Y(x_1,\ x_2)\equiv 1,$$

$$f(x_1,\ x_2)=1-0{,}5\ln\frac{(1+x_1^2+x_2^2)}{1+x_1^2+(x_2-1)^2}(\cos x_1^2-\cos(x_1^2+1))$$

Задача 2.

$$K=10(x_2-t_2)\sin[\mu(x_1+t_1)+(x_2-t_2)^2], \qquad Y(x_1,\ x_2)\equiv 1,$$

$$f(x_1,\ x_2)=1+\frac{20}{\mu}\sin\frac{\mu}{2}(0{,}5-x_2)\sin\left(\frac{\mu}{2}+\mu x_1+0{,}5+x_2^2-x_2\right),$$

$$\mu=\text{const}.$$

В табл. 1 и 2 приведены результаты расчетов. Время в скобках соответствует алгоритму $GJ(h)$.

Через $\delta(N_1 \times N_2)$ обозначена соответствующая среднеквадратичная абсолютная ошибка в виде числа с плавающей точкой.

Как видим, на сетке $N_1 \times N_2$. (N_1 — число точек дискретизации по k_1, N_2 — по x_2) алгоритм $A(h)$ работает тем быстрее, чем больше произведение $N_1 N_2$ (и чем больше N_2 по отношению к N_1).

В задаче 2 при $\mu = 2\pi$ ядро K и решение Y являются периодическими гладкими функциями, и в табл. 2 при этом хорошо просматривается резкое увеличение точности вычислений. Отметим, что стабилизация ошибки $\delta(N_1, N_2)$ при увеличении N_1 и N_2 не является признаком неустойчивости счета, а обусловлена относительно небольшой разрядностью компьютера и соответствующим накоплением ошибок. Соответственно, густота сети $N_1 \times N_2$ в тестовых задачах 1 и 2 была ограничена оперативной памятью.

Таблица 1

Задача 1. **Ошибка $\delta(N_1, N_2)$ и время счета**

$N_1 \times N_2$	4×4	4×8	4×16	4×32	8×8	8×16	16×16
$G(h)$	4,79E−2	2,85E−2	2,11E−2	1,88E−2	2,44E−2	1,42E−2	1,32E−2
t (c)	0 (1)	3 (4)	14 (21)	89 (138)	15 (22)	98 (148)	703 (1050)
$A(h)$	4,58E−2	2,86E−2	2,08E−2	1,78E−2	2,46E−2	1,48E−2	1,28E−2
t (c)	1	5	16	62	26	97	670

Таблица 2

Задача 2. Ошибка $\delta(N_1, N_2)$

μ	$N_1 \times N_2$	4×4	4×8	4×16	8×8	8×16	16×16
π	$G(h)$	4,69E−1	3,4[illegible]E−1	[illegible]03E−1	2,26E−1	1,69E−1	1,13E−1
	$A(h)$	2,29E−1	1,28E−1	[illegible]88E−1	1,23E−1	6,66E−2	6,37E−2
2π	$G(h)$	2,79E−12	3,67E−12	4,16E−12	4,88E−12	6,85E−12	7,6[illegible]E−12
	$A(h)$	3,74E−12	2,87E−12	3,23E−12	4,14E−12	5,57E−12	6,52E−12

При дополнительной гладкости ядра **K** и правой части f алгоритмы высокой точности можно получить как применением экстраполяции по Ричардсону (например путем счета на сетках $(N_1 \times N_2)$ и $(2N_1 \times 2N_2)$), так и использованием в схеме (5)—(6) кубатурных формул повышенной точности (как в одномерном случае, см. ([6])).

В заключение отметим, что предлагаемый подход проще методов, приведенных в работах ([4, 5]).

Институт математики Академии наук Армении
Ереванский государственный университет

Հայաստանի ԳԱ թղթակից անդամ Հ. Բ. ՆԵՐՍԻՍՅԱՆ, Ս. Ա. ԾԻՐՈՒՆՅԱՆ

Մասնակի տյոպլիցյան կորիզով բազմաչափ ինտեգրալ հավասարումների լուծման արագ ալգորիթմ

Դիտարկվում է (1) հավասարումը, որի լուծման հայտնի մեթոդներն ապահովում են համեմատաբար ոչ մեծ ճշտություն (Մոնտե-Կառլոյի մեթոդ) կամ պահանջում են շատ գործողությունների կատարում (Գաուսի մեթոդ)։

Առաջարկվող մեթոդը ըստ 1 արգումենտի մասնակի տյոպլիցյան (մասնակի տարբերական) (2) կորիզներով հավասարման լուծման գործողությունների քանակը զգալիորեն իջեցնում է։ Միաժամանակ կրճատվում է պահանջվող հիշողության չափը։ Հաշվարկների արդյունքները բերված են աղյուսակներում։

ЛИТЕРАТУРА—ԳՐԱԿԱՆՈՒԹՅՈՒՆ

[1] *А. Ф. Верлань, В. С. Сизиков*, Интегральные уравнения: методы, алгоритмы, программы. Наукова думка, Киев, 1986. [2] *I. Gohberg. I. Koltracht*, Numer. Math., v. 47, п. 233—238 (1985). [3] *I. Gohberg. I. Koltracht, P. Lancaster*, Integral Equat and Oper. Theory, v. 10, p. 577—594 (1987). [4] *С. Н. Воеводина*. Вычислительные методы и программирование, вып. 24, МГУ, с. 19—24, 1975. [5] *В. В. Воеводин, А. Г. Свешников, Е. Е. Тыртышников*, Вестн. МГУ, выч. мат. и кибернетика, № 1, 1980. [6] *А. Б. Нерсесян*, ДАН СССР, т. 30, № 3, с. 727—732 (1984). [7] *А. Б. Нерсесян*, ДАН АрмССР, т. 89, № 4, с. 171—176 (1989).