Параллельный алгоритм решения систем линейных уравнений в рациональных числах

А.О. Лапаев

Аннотация. Рассматривается параллельный метод решения систем линейных уравнений в кольце целых чисел методом p-адического подъема. Приводятся результаты экспериментов на кластере МСЦ РАН.

Keywords. параллельная компьютерная алгебра, системы линейных уравнений, p-адический подъем.

1. Введение.

Пусть система линейных уравнений в кольце целых чисел определена невырожденной квадратной матрицей A размера $n \times n$ и столбцом свободных членов B. Тогда $AX = B$ – матричная запись этой системы линейных уравнений.

Будем находить решение данной системы методом p-адического подъема [1]. Для восстановления результата в поле рациональных чисел воспользуемся алгоритмом Ванга [1].

Рассмотрим алгоритм Ванга и алгоритм p-адического подъема.

1.1. Алгоритм Ванга

Пусть отображение $f : Q \longmapsto Z_p$ по правилу: $\frac{a}{b} \to (a \mod p) \cdot (b^{-1} \mod p)$. Пусть $s = \max\{|a|, |b|\}$. Для того, чтобы отображение f было биекцией, необходимо и достаточно выполнения условия $p > 2 \cdot s^2$. Для нахождение прообраза $u = f(\frac{a}{b})$ воспользуемся алгоритмом Ванга [1]:

Algorithm Vang

Ввод: простой модуль p и $u \in Z_p$

Вывод: пара целых чисел (a, b), такая, что $a \equiv ub \mod p$, $|a|, |b| < \sqrt{p/2}$

$(a_1, a_2) := (p, u); (v1, v2) := (0, 1);$

while TRUE **do**

Работа выполнена при поддержке программы «Развитие потенциала высшей школы» (проект 2.1.1/1853) .

if $v_2 \geq \sqrt{p/2}$ **then return** NIL;
if $a_2 < \sqrt{p/2}$ **then** $(sign(v_2) \cdot a_2, |v_2|)$;
$q := \lfloor a_1/a_2 \rfloor$; $(a_1, v_1) := (a_1, v_1) - q(a_2, v_2)$;
ПЕРЕСТАНОВКА$((a_1, v_1) \leftrightarrow (a2, v_2))$
end while

1.2. Алгоритм p-адического подъема

Алгоритм p-адического подъема [1] позволяет найти решение системы линейных уравнений по модулю p^k, если известно решение по модулю p^{k-1} и A_p^{-1} – матрица, обратная к A по модулю p, где p–простое число, $k > 1$. Пусть $H(A) = \sqrt{\prod_{j=1}^{j=n} \sum_{i=1}^{i=n} a_{ij}^2}$ – оценка Адамара определителя матрицы A.

Algorithm p-adic lifting

Ввод: Матрицы A, A^{-1}, столбец свободных членов B.
Вывод: Решение системы x по модулю $l > 2 \cdot H^2(A|B)$
$l := 1; x := 0; c^* := B$;
repeat
$\hat{x} := (A^{-1}c^*) \mod p$;
$c^* := (c^* - A\hat{x})/p$;
$x := x + l\hat{x}$;
$l := lp$;
until $l < 2 \cdot H^2(A) + 1$;

1.3. Алгоритм решения систем линейных уравнений методом p-адического подъема

Решение системы будем находить следующим образом:

1. Выберем p – простое число, такое, что $\det A$ не является кратным числу p.
2. Вычислим A_p^{-1} - матрицу, обратную к A по простому модулю p.
3. С помощью алгоритма p-адического подъема найдем решение системы по модулю p^k. При этом подъем будем продолжать до тех пор, пока $p^k < 2 \cdot H^2(A) + 1$.
4. При помощи алгоритма Ванга восстановим вектор решений X в поле Q.

2. Параллельный алгоритм

Пусть ЭВМ содержит $m = 4^M$ процессоров с номерами $0, 1, \ldots, m-1$. Обозначим $s = \sqrt{m}$. Представим множество процессоров в виде квадратной решетки, причем процессор с номером k будет иметь в решетке индексы $(i, j) = (\lfloor k/s \rfloor, k \mod s)$. Пусть размер матрицы $n = 2^N, n \geq m$. Разобьем матрицу A и A_p^{-1} на m квадратных блоков размерами $\frac{n}{s} \times \frac{n}{s}$. Обозначим A_{ij}, $A_{ij_p}^{-1}$ – блоки матриц A и A_p^{-1} с номерами (i, j) соответственно. Пусть на процессоре с номером

$is + j$ находятся A_{ij}, $A^{-1}_{ij_p}$. Приведем параллельный алгоритм p-адического подъема.

1. Процессор с номером 0 выполняет разбиение вектора B на s равных частей B_i. Выполняется рассылка блока B_i процессорам с номерами $is + j, j = 0, \ldots, s - 1$.

   Шаги алгоритма с 2 по 17 выполняются на каждом процессоре параллельно.
2. $C := A^{-1}_{ij_p} B_i \mod p$.
3. Выполняется разбиение C на s частей C_t, $t = 0, \ldots, s$.
4. Часть C_t посылается на процессор с номером $is + j$, где i – строчный номер данного процессора, $t = 0, \ldots, s - 1$
5. Все полученные блоки C_t суммируются и результат записывается в $\hat{C}$.
6. Выполняется рассылка $\hat{C}$ процессорам с номерами $js+i$, где i – строчный номер данного процессора, $j = 0, \ldots, s - 1$
7. Из полученых блоков составляется вектор $X := (\hat{C}_0, \hat{C}_1, \ldots, \hat{C}_{(m-1)})$, где $\hat{C}_i$ – часть вектора, полученная от процессора с номером столбца i.
8. $\hat{X} := \hat{X} + lX$, $l = lp$.
9. $B_i := (B_i - A_{ij}X)/p$.
10. B_i разбивается на s частей B_{i_t}, $t = 0, \ldots, s - 1$.
11. Выполняется посылка B_{i_t} на процессоры с номерами $is+t$, где i – строчный номер данного процессора, $j = 0, \ldots, s - 1$.
12. Выполняется суммирование всех полученных блоков B_{i_t} и результат записывается в $\hat{C}$.
13. Выполняется рассылка $\hat{C}$ процессорам с номерами $js + i$, где где i – строчный номер данного процессора.
14. Из полученых блоков составляется вектор $B_i := (\hat{C}_0, \hat{C}_1, \ldots, \hat{C}_{(m-1)})$, где $\hat{C}_i$ – часть вектора, полученная от процессора с координатами $(\cdot, i)$.
15. Если $l < 2 \cdot H^2(A) + 1$, то переход к шагу 2.
16. Выполняется алгоритм Ванга для элементов вектора X с индексами $\frac{in}{m}, \ldots, \frac{(i+1)n}{m} - 1$.
17. Восстановленный алгоритмом Ванга фрагмент вектора X отправляется процессору с номером 0.
18. Процессор с номером 0 составляет вектор решения из частей, полученных от каждого из процессоров.
19. Конец вычислений.

По приведенному выше алгоритму был разработан программный комплекс. Результаты экспериментов на кластере МСЦ РАН будут представлены в докладе.

Список литературы

[1] Malaschonok G.I. Solution of Systems of Linear Equations by the p-Adic Method. Programming and Computer Software, Vol. 29, No. 2, 2003, pp. 59–71.

[2] Малашонок Г.И. Матричные методы вычислений в коммутативных кольцах. Тамбов, 2002.

А.О. Лапаев
ул. Интернациональная, 33
Тамбов
Россия

e-mail: alapaev@gmail.com