

ИНДЕКС 3649

Препринт ЕФН-1224(10)-90

ԵՐԵՎԱՆԻ ՖԻԶԻԿԱՅԻ ԻՆՍՏԻՏՈՒՏ  
ЕРЕВАНСКИЙ ФИЗИЧЕСКИЙ ИНСТИТУТ  
YEREVAN PHYSICS INSTITUTE

---

М.А.МХИТАРЯН, А.Б.НЕРСЕСЯН

ПАРАЛЛЕЛЬНО-ПОТОЧНАЯ ИНТЕРПРЕТАЦИЯ  
НЕКОТОРЫХ АЛГОРИТМОВ ТРЕУГОЛЬНОГО  
РАЗЛОЖЕНИЯ ИНТЕГРАЛЬНЫХ ОПЕРАТОРОВ  
ВТОРОГО РОДА



ЕРЕВАНСКИЙ ФИЗИЧЕСКИЙ ИНСТИТУТ

ЦНИИАтоминформ  
ЕРЕВАН - 1989

Մ.Ա.ՄԴԻԹԱՐՅԱՆ, Ա.Բ.ՆԵՐՍԻՍՅԱՆ

ԵՐԿՐՈՐԴ ԿԱՐԳԻ ԻՆՏԵԳՐԱԼ ՕՊԵՐԱՏՈՐՆԵՐԻ ԵՌԱՆԿՑՈՒ-  
ՆԱՅԻՆ ՏՐՈՂՄԱՆ ՈՐՈՇ ԱԼԳՈՐԻԹՄՆԵՐԻ ԳՋՈՒԳԱՀԵՌ-  
ՀՈՍՔԱՅԻՆ ՄԵԿՆԱՐԱՆՈՒՄԸ

Առաջարկված են սխտուրիկ զանգվածներ եռանկյունային տրոհման ո-  
րոշ ալգորիթմների իրագործման համար: Յուրաքանչյուր կառուցվածքի հա-  
մար որոշվում է հաշվարկի արդյունավետության գործակիցը, որը թույլ  
է տալիս գնահատել սարքավորումների /պրոցեսորային մասերի/ կիրառման  
աստիճանը:

Երևանի ֆիզիկայի ինստիտուտ

Երևան 1990

Решение интегрального уравнения второго рода с гладким яд-  
ром традиционно (см., например, [1]) сводят к задаче решения  
алгебраической системы. Решение же последней может быть осуще-  
ствлено на основе богатого арсенала методов. В частности, ме-  
тод Гаусса и его разновидности допускают эффективное распа-  
раллеливание и соответствующую параллельно-поточную реализацию  
на систолических массивах (см. [2], [3]).

В работе [4] были предложены новые методы решения интеграль-  
ных уравнений. В данной работе мы ограничимся соответствующими  
алгоритмами треугольного разложения интегральных операторов  
второго рода. Их высокая операционная сложность делает невоз-  
можным решение задачи на последовательных ЭВМ за приемлемое вре-  
мя. Появление сверхбольших интегральных схем (СБИС) делает тех-  
нически возможным построение параллельных вычислительных систем,  
содержащих очень большое число процессорных элементов. Вместе  
с тем, СБИС - технология предъявляет ряд требований к алгорит-

мам для их успешной реализации в аппаратуре (см. [3]).

С учетом пространственно-временных особенностей реализации приведенных алгоритмов с различной степенью точности вычисления в работе предложены соответствующие систолические массивы с необходимой коммуникацией. Для каждой структуры определяется коэффициент эффективности вычислений, позволяющий оценить степень использования оборудования (процессорных элементов).

I. Пусть  $R(i, j, n) \stackrel{\text{def}}{=} R(ih, jh, nh)$  ( $i, j, n \in \mathbb{Z}$ ) т.е. формально процесс дискретизации можно рассматривать на сетке с узлами, имеющими целочисленные координаты. Учитывая такую формализацию, процесс (см. [4]) можно описать следующим образом:

(входные данные (начальное присваивание))

```
for all  $0 < i, j < N$  do  $R(i, j, 0) = K(i, j)$ ,
```

(внутренние вычисления)

```
for  $n = 0$  to  $N - 1$ 
```

```
for all  $n + 1 < i, j < N$  do
```

```
begin
```

```
(I)  $R(i, j, n + 1) = R(i, j, n) + hR(i, n, n) \cdot R(n, j, n)$ ,
```

```
end  
end  
end
```

(выходные вычисления)

```
for all  $0 < n < N$ 
```

```
for  $i = n, n < j < N$  do  $\beta_{nj} = R(n, j, n)$ 
```

```
for  $j = n, n < i < N$  do  $\alpha_{in} = R(i, n, n)$ .
```

На каждом этапе выражение  $\alpha'_{in} = h \cdot R(i, n, n)$  вычисляется  $(N+1-i)$  раз, т.е. вкладом  $\sum_{i=1}^N (N+1-i) = O(N^2)$  в общее число операций можно пренебречь. Таким образом, последнее будет равно  $\Omega = O(N^3/3)$ . За единицу времени принимается время вычисления составного арифметического выражения  $S \leftarrow S + \alpha \cdot \beta$ . Заметим, что количество операций при реализации алгоритма, описываемого формулой (I), совпадает с операционной сложностью известного алгоритма решения системы линейных уравнений методом Гаусса.

Пусть имеется ортогональная сеть, состоящая из  $(N+1)^2$  - вычислителей (рис. I,  $N = 3$ ). В памяти  $(i, j)$  - процессора хранится  $R(i, j, 0)$ , а при  $i > j$  - дополнительно  $h$ . Реализация алгоритма начинается с  $(0, 0)$  - процессора, в момент времени  $t = 1$  отправляющего в  $(0, 1)$  и  $(1, 0)$  сигнал возбуждения (можно послать единицу в качестве информационного импульса). А в следующий момент времени ( $t = 2$ )  $(0, 1)$  - вычислитель посредством регистра  $R_2$  отправляет вертикально вниз  $R(0, 1, 0)$   $(1, 1)$  - процессору, а  $(0, 2)$  - процессору сигнал возбуждения;  $(1, 0)$  - вычислитель считает  $\alpha'_{10} = hR(1, 0, 0)$  и с помощью регистра  $R_1$  транслирует  $\alpha'_{10}$  в  $(1, 1)$  - процессор, а в  $(2, 0)$  - единицу для возбуждения. При  $t = 3$  в  $(1, 1)$  - процессоре вычисляется  $R(1, 1, 1)$ ;  $(0, 2)$  - процессор транслирует в  $(1, 2)$  - процессор  $R(0, 2, 0)$ , а в  $(0, 3)$  - вычислитель - сигнал возбуждения;  $(2, 0)$  - процессор вычисляет  $\alpha'_{20} = hR(2, 0, 0)$  и передает  $\alpha'_{20}$   $(2, 1)$  - процессору, а в  $(3, 0)$  - процессор направляет информационный импульс и т.д. Таким образом, на первом этапе ( $n = 0$ ) вычисляются  $R(i, j, 1)$ .

где  $i, j \geq 1$ .

$(n+1)$ -ый этап начинается с диагонального элемента  $R(n+1, n+1, n+1)$  на третьем шаге после вычисления  $R(n, n, n)$ , и на  $(n+1)$ -ом этапе обрабатываются элементы для которых  $i, j \geq n+1$ .  $R(i, n+1, n+1)$  и  $R(n+1, j, n+1)$  запоминаются до конца вычислений в локальной памяти  $(i, n+1)$  ( $i \geq n+1$ ) и  $(n+1, j)$  ( $j > n+1$ ) вычислителей соответственно. А промежуточные значения  $R(i, j, n+1)$  ( $i, j > n+1$ ) хранятся в локальной памяти  $(i, j)$ -ого вычислителя до следующего этапа.

$R(i, j, n+1)$  вычисляется в момент времени  $t_{ij} = n+1+i+j$ . В ортогональной сети по горизонтали вправо через регистр  $R_1$  распространяется  $\alpha'_{in} = hR(i, n, n)$  ( $i \neq 0$ ), а по вертикали вниз (посредством регистра  $R_2$ )  $R(n, j, n)$  ( $j \neq 0$ ). Процесс заканчивается в  $(N, N)$ -процессоре вычислением  $R(N, N, N)$ . Время параллельно-поточной реализации на ортогональной сети алгоритма по формуле [1] составит величину

$$T_{(n+1)^2} = \max_{\substack{0 \leq i \leq n-1 \\ n+1 \leq i, j \leq N}} (n+1+i+j) = 3N$$

коэффициент эффективности  $K_{эф} = \frac{1}{9}$ .

2. Интерес представляет также вычисление  $R(i, j, n+1)$  по другой формуле (при существовании вторых производных у ядра  $K(i, j)$ ) с более высокой точностью конечного результата (см. [4]). Пусть  $R(i, j, n) \stackrel{\text{def}}{=} R((i + \frac{1}{2})h, (j + \frac{1}{2})h, nh)$ . Схема вышеуказанного алгоритма такова:

(входные данные (начальное присваивание))

for all  $0 \leq i, j \leq N-1$  do  $R(i, j, 0) = K(i, j)$ ,

(внутренние вычисления)

for  $n=0$  to  $N-2$

for all  $n+1 \leq i, j \leq N-1$  do

begin

$$(2) \quad R(i, j, n+1) = R(i, j, n) + \frac{R(i, n, n) \cdot R(n, j, n)}{(\frac{1}{h} - R(n, n, n))},$$

end

end

end

(выходные вычисления (финальное присваивание))

for all  $0 \leq n \leq N-1$

for  $i=n, n \leq j \leq N$  do  $\beta_{nj} = R(n, j, n)$ .

for  $j=n, n \leq i \leq N$  do  $\alpha_{in} = R(i, n, n)$ .

Как и в предыдущем случае,  $\Omega = O(N^3/3)$ , с учетом того факта, что вычисление  $\alpha'_{in} = \frac{R(i, n, n)}{(\frac{1}{h} - R(n, n, n))}$  на каждом этапе производится  $(N-i)$  раз.

В последнем случае вычислительная сеть аналогична первой, т.е. сеть из  $N^2$  ортогонально связанных процессоров реализует данные вычисления. В локальной памяти  $(i, j)$  - вычислителя хранится  $R(i, j, 0)$ , а в  $(i, i)$  ( $i \neq N$ ) - дополнительно  $\frac{1}{h}$ .

Процесс в вычислительной сети протекает аналогично описанному в первом случае, за исключением того, что диагональный процессор  $(n, n)$  на  $(n+1)$ -ом этапе вычисляет  $(\frac{1}{h} \cdot R(n, n, n))$ , транслирует в  $(n+1, n)$  - процессор, в котором в следующий момент времени вычисляется  $\alpha_{n+1, n}$ . Величина  $R(i, j, n+1)$  высчитывается в момент времени  $t_{ij} = n+1+i+j$ . Время реализации данного алгоритма:

$$T_{N^2} = \max_{\substack{0 \leq n \leq N-2 \\ n+1 \leq i, j \leq N-1}} (n+1+i+j) = 3(N-1)$$

Коэффициент эффективности  $K_{эф} = \frac{1}{9}$ .

С такой же точностью - порядка  $O(h^2) - R(i, j, n+1) (i, j \geq n+1)$  можно вычислить по другой формуле

$$(3) \quad R(i, j, n+1) = R(i, j, n) + R(i, n, n) \cdot \beta_{nj} + \alpha_{in+1} \cdot f_{nj}, \text{ где}$$

введены соответствующие обозначения:

$$\alpha_{nn} = \frac{h}{2} + \frac{h^2}{2} R(n, n, n)$$

$$\beta_{nj} = \alpha_{nn} \cdot R(n, j, n)$$

$$\alpha_{n+1, n} = \frac{h}{2} R(n+1, n, n)$$

$$\alpha_{in+1} = \frac{h}{2} R(i, n+1, n)$$

$$f_{nj} = R(n+1, j, n) + R(n, j, n) \cdot \alpha_{n+1, n}$$

$\Omega = O(\frac{2}{3} N^3)$  (вкладом операций, необходимых для подсчета коэффициентов, в операционную сложность алгоритма можно пренебречь).

Этот процесс можно реализовать с помощью вычислительной сети (количество процессоров -  $(N+1)^2$ ), изображенной на рис.2 ( $N = 3$ ), на которой представлены также типы процессоров.

В локальной памяти  $(i, j)$  - процессора хранится  $R(i, j, 0)$ , в  $(0, 0)$  и  $(1, 0)$  - процессорах, а также в  $(i, j)$  - вычислителе  $(i \geq j, j \neq 0) \frac{h}{2}$ , а в  $(i, i)$  - процессоре  $(i \neq N) \frac{h^2}{2}$ . Величина  $R(i, j, n+1) (i, j \geq n+1)$  вычисляется за два такта. Значение  $R(i, j, n+1)$  зависит от коэффициента  $\alpha_{in+1} (i \geq n+1)$  вычисляемого и задерживаемого в нем на один такт. Таким образом, промежуточное значение при вычислении  $R(i, j, n+1)$  запоминается  $(i, j)$  - процессором в течение двух тактов. А относительно величин  $R(i, n+1, n+1) (i \geq n+1)$ ,  $R(n+1, j, n+1) (j > n+1)$ ,  $R(i, j, n+1) (i, j > n+1)$  замечания те же, что и в предыдущих случаях. Время реализации  $T_{(N+1)^2} = 5N$ , а коэффициент эффективности  $K_{эф} = \frac{1}{15}$ .

По сравнению с предыдущим алгоритмом очевидно, что вычислительная сеть, реализующая данные вычисления, отличается более сложной коммуникацией и размерами локальной памяти каждого процессора при операционной сложности, вдвое превосходящей количество операций второго алгоритма. Таким образом, при одинаковой точности вычислений предпочтительнее использовать при параллельно-поточной реализации схему, изображенную на рис.1. Этот пример также показывает, что не каждый алгоритм данной точности может быть рационально систолизирован.

3. Перейдем к рассмотрению последней формулы, описывающей тот же процесс, но с более высокой степенью точности нахождения  $R(i, j, n+1)$  - порядка  $O(h^4)$ . Необходимо отметить, что  $i, j \in \{\frac{K}{2}\}$ , где  $K = \overline{0, 2N}$  и  $0 \leq n \leq N-1, n \in \mathbb{Z}$ . Пусть

$$R(i, j, n) \stackrel{\text{def}}{=} R(ih, jh, nh).$$

Структура алгоритма следующая:

(входные данные (начальное присваивание))

for all  $0 < i, j < N$  do  $R(i, j, 0) = K(i, j)$ ,

(внутренние вычисления)

for  $n = 0$  to  $N-1$

for all  $n+1 < i, j < N$  do

begin

(4)

$$R(i, j, n+1) = R(i, j, n) + \frac{h}{6} R(i, n, n) \cdot R(n, j, n) + \frac{h}{6} R(i, n+1, n) \times$$

$$\times R(n+1, j, n) + \frac{2h}{3} R(i, n+\frac{1}{2}, n) \cdot R(n+\frac{1}{2}, j, n) + R(i, n+\frac{1}{2}, n) \times$$

$$\times \frac{R(n+\frac{1}{2}, n+\frac{1}{2}, n)}{\frac{1}{h^2} - R^2(n+\frac{1}{2}, n+\frac{1}{2}, n)} \cdot \left\{ R(n+\frac{1}{2}, j, n) \cdot \left(1 + \frac{2h}{3} R(n+\frac{1}{2}, n+\frac{1}{2}, n)\right) + \right.$$

$$\left. + \frac{h}{6} R(n+\frac{1}{2}, n, n) \cdot R(n, j, n) + \frac{h}{6} R(n+\frac{1}{2}, n+1, n) \cdot R(n+1, j, n) \right\},$$

end

end

end

(выходные вычисления)

for all  $0 < n < N$

for  $i = n, n < j < N$  do  $\eta_{nj} = R(n, j, n)$

for  $j = n, n < i < N$  do  $\xi_{in} = R(i, n, n)$

for all  $0 < n < N-1$

for  $i = n + \frac{1}{2}, n < j < N$  do  $\eta_{n+\frac{1}{2}, j} = R(n + \frac{1}{2}, j, n)$

for  $j = n + \frac{1}{2}, n < i < N$  do  $\xi_{i, n+\frac{1}{2}} = R(i, n + \frac{1}{2}, n)$ .

Формулу (4) можно записать в упрощенном виде, если ввести следующие обозначения:

$$d'_{n+\frac{1}{2}, n+\frac{1}{2}} = \frac{R(n+\frac{1}{2}, n+\frac{1}{2}, n)}{\frac{1}{h^2} - R^2(n+\frac{1}{2}, n+\frac{1}{2}, n)}$$

$$b'_{n+\frac{1}{2}, n+\frac{1}{2}} = 1 + \frac{2h}{3} R(n+\frac{1}{2}, n+\frac{1}{2}, n)$$

$$c'_{n+\frac{1}{2}, h} = \frac{h}{6} R(n+\frac{1}{2}, n, n)$$

$$d'_{n+\frac{1}{2}, n+1} = \frac{h}{6} R(n+\frac{1}{2}, n+1, n)$$

$$e_{in} = \frac{h}{6} R(i, n, n)$$

$$e_{i, n+\frac{1}{2}} = \frac{2h}{3} R(i, n+\frac{1}{2}, n)$$

$$z_{i, n+\frac{1}{2}} = d'_{n+\frac{1}{2}, n+\frac{1}{2}} \cdot R(i, n+\frac{1}{2}, n)$$

$$\omega_{in+1} = \frac{h}{6} R(i, n+1, n)$$

$$\delta_{n+1, j} = c'_{n+\frac{1}{2}, n} R(n, j, n) + b'_{n+\frac{1}{2}, h+\frac{1}{2}} \cdot R(n+\frac{1}{2}, j, n) + d'_{n+\frac{1}{2}, n+1} \cdot R(n+1, j, n)$$

Таким образом, имеем:

$$R(i, j, n+1) = R(i, j, n) + \varepsilon_{in} \cdot R(n, j, n) + \varepsilon_{i, n+\frac{1}{2}} \cdot R(n+\frac{1}{2}, j, n) + \omega_{i, n+1} \cdot R(n+1, j, n) + z_{i, n+\frac{1}{2}} \cdot \delta_{n+1, j} \quad (5)$$

Операционная сложность данного алгоритма  $\Omega = O\left(\frac{16}{3} N^3\right)$  (без учета операций, необходимых для вычисления коэффициентов, как и в предыдущих случаях).

Вычислительная сеть из  $(N+1)^2$  процессоров для реализации данного алгоритма изображена на рис.3 ( $N = 3$ ).

В локальной памяти  $(i, j)$  - процессора, где  $i, j \in Z$  в начальный момент времени хранятся следующие элементы:  $R(i, j, 0)$ ,  $R(i+\frac{1}{2}, j, 0)$ ,  $R(i, j+\frac{1}{2}, 0)$ ,  $R(i+\frac{1}{2}, j+\frac{1}{2}, 0)$  ( $i \neq N, j \neq N$ );  $R(N, j, 0)$  и  $R(N, j+\frac{1}{2}, 0)$  ( $i = N, j \neq N$ );  $R(i, N, 0)$ ,  $R(i+\frac{1}{2}, N, 0)$  ( $j = N, i \neq N$ );  $R(N, N, 0)$  ( $i = j = N$ ).

Дополнительно  $(n, n)$  - вычислитель запоминает  $\frac{1}{h^2}$ ,  $\frac{2h}{3}$ ,  $\frac{h}{6}$ ,  $(n, n+1)$  - процессор  $\frac{h}{6}$ ;  $(i, n)$  - вычислитель  $\frac{2h}{3}$ ,  $\frac{h}{6}$ ;  $(i, n+1)$  - вычислитель  $\frac{h}{6}$ . На рис.4 показана реализация процессора при  $N = 2$ . Последовательность обработки элементов в блоке  $(i, j)$  такова:  $(i, j)$ ,  $(i+\frac{1}{2}, j)$ ,  $(i, j+\frac{1}{2})$ ,  $(i+\frac{1}{2}, j+\frac{1}{2})$ , а очередность выполнения операций при вычислении коэффициентов и  $R(i, j, n+1)$  по формуле (5) соответствует записи.

Обозначение (см.рис.4)  $\textcircled{1}$   $A$   $\textcircled{2}$  означает, что в момент времени  $t_1$  в процессоре выполняется  $t_2$ -ая операция при вычислении величины  $A$  (если  $t_2$  не указано, то  $t_2 = 1$ ), а обозначение на стрелке  $\boxed{t_1}$   $A$  означает, что в момент времени  $t_1$   $A$  транслируется в соответствующий вычислитель.

Промежуточные результаты при вычислении  $R(i, j, n+1)$  и коэффициентов  $\alpha'_{n+\frac{1}{2}, n+\frac{1}{2}}$ ,  $\delta_{n+1, j}$  запоминаются соответствующими процессорами в течение необходимого времени (см.рис.4). Коэффициенты  $\varepsilon_{in}$ ,  $\varepsilon_{i, n+\frac{1}{2}}$ ,  $z_{i, n+\frac{1}{2}}$ ,  $\omega_{i, n+1}$  хранятся в локальной памяти  $(i, j)$  - блока  $(i, j \in Z, i, j > n+1)$  на  $(n+1)$  - ом этапе в течение времени, пока данный процессор обрабатывает соответствующие элементы.

Коэффициент  $\alpha'_{n+\frac{1}{2}, n+1}$  задерживается  $(n, n+1)$  - вычислителем на один такт, а  $\delta_{n+1, j}$  -  $(n, j)$  - процессором также на один такт.

Элементы, обозначенные в структуре алгоритма как финальные, запоминаются в локальной памяти соответствующих процессоров до конца процесса, а  $R(i, j, n+1)$  ( $i, j > n+1$ ) хранится до следующего этапа в памяти  $(i, j)$  - вычислителя ( $i, j > n+1$ ).

Величина  $R(i, j, n+1)$  ( $i, j \in Z$ ) будет вычислена в момент времени  $t_{ij} = i+j+5+18n$ ;  $R(i+\frac{1}{2}, j, n+1)$  в  $t_{i+\frac{1}{2}, j} = t_{ij}+5$ ;

$R(i, j+\frac{1}{2}, n+1)$  в момент времени  $t_{i, j+\frac{1}{2}} = t_{ij}+9$ ;  $R(i+\frac{1}{2}, j+\frac{1}{2}, n+1)$  в момент времени  $t_{i+\frac{1}{2}, j+\frac{1}{2}} = t_{ij}+13$ . Процесс заканчивается в  $(N, N)$  - вычислителе.

$$T_{(N+1)^2} = \max_{\substack{0 < n < N-1 \\ n+1 \leq i, j < N}} (i+j+5+18n) = 20N-13$$

$$\text{Коэффициент эффективности } K_{\text{эф}} = \frac{4}{15}$$

В заключение отметим, что изучаемые выше алгоритмы относятся к построению приближенного треугольного разложения интегрального оператора, которое можно использовать при решении конкретных

уравнений. Разумеется, в процессе решения также можно поставить вопрос о систолизации, но здесь на этом останавливаться не будем. Развитие такого подхода - в применении к более точным алгоритмам работы [4] - позволит строить эффективные систолические массивы в случае уравнений с ядрами повышенной гладкости.

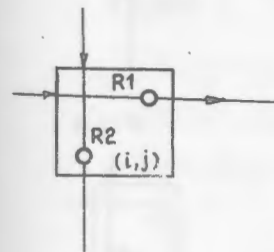
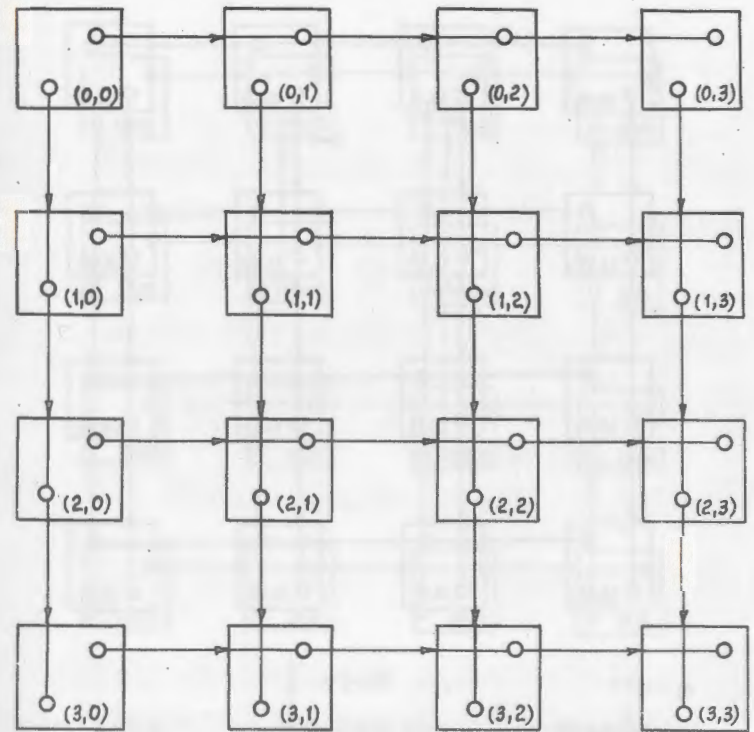
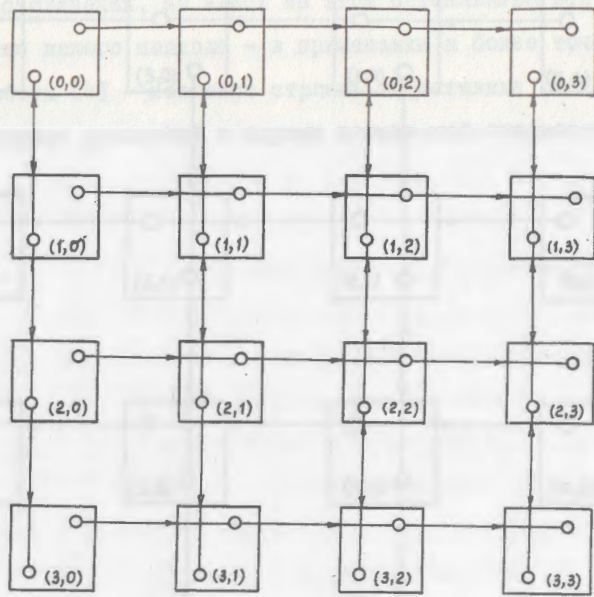
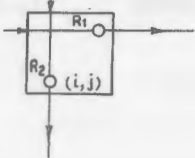


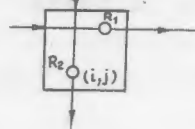
Рис. I



a)  $i > j + 1$



б)  $i = j + 1$



в)  $i < j$

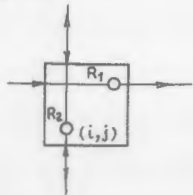
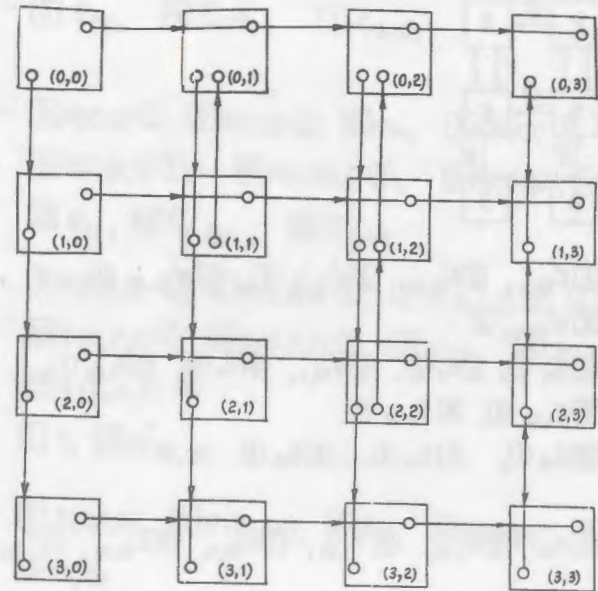
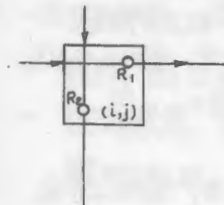


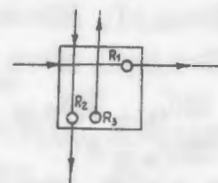
Рис. 2



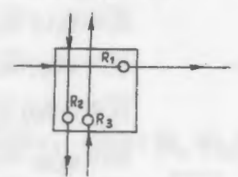
a)  $i > j; i = j = 0$



б)  $i = j (\neq 0, \neq N)$



в)  $i < j (j \neq N)$



г)  $j = N$

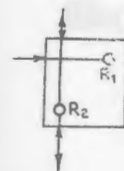
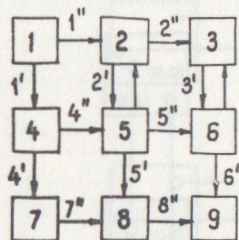


Рис. 3



- 1** - **1**  $C'_{\frac{1}{2},0}$ , **2**  $B'_{\frac{1}{2},\frac{1}{2}}$ , **3**  $d'_{\frac{1}{2},\frac{1}{2}}$  ①, **4**  $d'_{\frac{1}{2},\frac{1}{2}}$  ②,  
**5**  $d'_{\frac{1}{2},\frac{1}{2}}$  ③  
**2** - **2**  $\delta_{11}$  ①, **3**  $\delta_{11}$  ②, **4**  $d'_{\frac{1}{2},1}$ , **5**  $\delta_{11}$  ③, **12**  $\delta_{1,\frac{3}{2}}$  ①,  
**10**  $\delta_{1,\frac{3}{2}}$  ②, **14**  $\delta_{1,\frac{3}{2}}$  ③  
**3** - **3**  $\delta_{12}$  ①, **4**  $\delta_{12}$  ②, **6**  $\delta_{12}$  ③  
**4** - **2**  $e_{10}$ , **3**  $\epsilon_{1,\frac{1}{2}}$ , **6**  $Z_{1,\frac{1}{2}}$ , **7**  $e_{\frac{3}{2},0}$ , **8**  $\epsilon_{\frac{3}{2},\frac{1}{2}}$ , **11**  $Z_{\frac{3}{2},\frac{1}{2}}$   
**5** - **3**  $R(1,1,1)$  ①, **4**  $R(1,1,1)$  ②, **5**  $\omega_{11}$ , **6**  $R(1,1,1)$  ③, **7**  $R(1,1,1)$  ④,  
**8**  $R(\frac{3}{2},1)$  ①, **9**  $R(\frac{3}{2},1)$  ②, **10**  $\omega_{\frac{3}{2},1}$ , **11**  $R(\frac{3}{2},1)$  ③,  
**12**  $R(\frac{3}{2},1)$  ④, **13**  $R(1,\frac{3}{2},1)$  ①, **14**  $R(1,\frac{3}{2},1)$  ②, **15**  $R(1,\frac{3}{2},1)$  ③,  
**16**  $R(1,\frac{3}{2},1)$  ④, **17**  $R(\frac{3}{2},\frac{3}{2},1)$  ①, **18**  $R(\frac{3}{2},\frac{3}{2},1)$  ②, **19**  $R(\frac{3}{2},\frac{3}{2},1)$  ③,  
**20**  $R(\frac{3}{2},\frac{3}{2},1)$  ④, **21**  $C'_{\frac{3}{2},1}$ , **22**  $B'_{\frac{3}{2},\frac{3}{2}}$ , **23**  $d'_{\frac{3}{2},\frac{3}{2}}$  ①,  
**24**  $d'_{\frac{3}{2},\frac{3}{2}}$  ②, **25**  $d'_{\frac{3}{2},\frac{3}{2}}$  ③  
**6** - **4**  $R(1,2,1)$  ①, **5**  $R(1,2,1)$  ②, **7**  $R(1,2,1)$  ③, **8**  $R(1,2,1)$  ④,  
**9**  $R(\frac{3}{2},2,1)$  ①, **10**  $R(\frac{3}{2},2,1)$  ②, **12**  $R(\frac{3}{2},2,1)$  ③, **13**  $R(\frac{3}{2},2,1)$  ④,  
**22**  $\delta_{22}$  ①, **23**  $\delta_{22}$  ②, **24**  $d'_{\frac{3}{2},2}$ , **25**  $\delta_{22}$  ③

Рис. 4

**7** - **3**  $e_{20}$ , **4**  $\epsilon_{2,\frac{1}{2}}$ , **7**  $Z_{2,\frac{1}{2}}$

**8** - **4**  $R(2,1,1)$  ①, **5**  $R(2,1,1)$  ②, **6**  $\omega_{21}$ , **7**  $R(2,1,1)$  ③, **8**  $R(2,1,1)$  ④,  
**14**  $R(2,\frac{3}{2},1)$  ①, **15**  $R(2,\frac{3}{2},1)$  ②, **16**  $R(2,\frac{3}{2},1)$  ③, **17**  $R(2,\frac{3}{2},1)$  ④,  
**22**  $e_{21}$ , **23**  $\epsilon_{2,\frac{3}{2}}$ , **26**  $Z_{2,\frac{3}{2}}$

**9** - **5**  $R(2,2,1)$  ①, **6**  $R(2,2,1)$  ②, **8**  $R(2,2,1)$  ③, **9**  $R(2,2,1)$  ④,  
**23**  $R(2,2,2)$  ①, **24**  $R(2,2,2)$  ②, **25**  $\omega_{22}$ , **26**  $R(2,2,2)$  ③,  
**27**  $R(2,2,2)$  ④.

$1' \uparrow$  - **1** 1, **5**  $d'_{\frac{1}{2},\frac{1}{2}}$

$2' \uparrow$  - **2**  $R(0,1,0)$ , **3**  $R(\frac{1}{2},1,0)$ , **6**  $\delta_{11}$ , **12**  $R(0,\frac{3}{2},0)$ , **13**  $R(\frac{1}{2},\frac{3}{2},0)$ ,  
**15**  $\delta_{1,\frac{3}{2}}$

$2' \uparrow$  - **4**  $R(1,1,0)$ , **13**  $R(1,\frac{3}{2},0)$

$3' \uparrow$  - **3**  $R(0,2,0)$ , **4**  $R(\frac{1}{2},2,0)$ , **7**  $\delta_{12}$

$3' \uparrow$  - **5**  $R(1,2,0)$

$4' \uparrow$  - **2** 1, **6**  $d'_{\frac{1}{2},\frac{1}{2}}$

$5' \uparrow$  - **3**  $R(0,1,0)$ , **4**  $R(\frac{1}{2},1,0)$ , **6**  $R(1,1,0)$ , **7**  $\delta_{11}$ , **14**  $R(\frac{1}{2},\frac{3}{2},0)$ ,  
**15**  $R(1,\frac{3}{2},0)$ , **16**  $\delta_{1,\frac{3}{2}}$ , **25**  $d'_{\frac{3}{2},\frac{3}{2}}$

$6' \uparrow$  - **4**  $R(0,2,0)$ , **5**  $R(\frac{1}{2},2,0)$ , **7**  $R(1,2,0)$ , **8**  $\delta_{12}$ , **22**  $R(1,2,1)$ ,  
**23**  $R(\frac{3}{2},2,1)$ , **24**  $R(2,2,1)$ , **28**  $\delta_{22}$

Продолжение рис. 4

$$\begin{aligned}
1'' & - \boxed{1} C'_{1/2,0}, \boxed{2} B'_{1/2,1/2} \\
2'' & - \boxed{2} C'_{1/2,0}, \boxed{5} d'_{1/2,1} \\
4'' & - \boxed{2} e_{10}, \boxed{3} \varepsilon_{1,1/2}, \boxed{6} z_{1,1/2}, \boxed{7} e_{3/2,0}, \boxed{8} \varepsilon_{3/2,1/2}, \boxed{11} z_{3/2,1/2} \\
5'' & - \boxed{3} e_{10}, \boxed{4} \varepsilon_{1,1/2}, \boxed{6} \omega_{11}, \boxed{7} z_{1,1/2}, \boxed{8} e_{3/2,0}, \boxed{9} \varepsilon_{3/2,1/2}, \\
& \boxed{11} \omega_{3/2,1}, \boxed{12} z_{3/2,1/2}, \boxed{21} e'_{3/2,1}, \boxed{22} B'_{3/2}, \\
7'' & - \boxed{3} e_{20}, \boxed{4} \varepsilon_{2,1/2}, \boxed{7} z_{2,1/2} \\
8'' & - \boxed{4} e_{20}, \boxed{5} \varepsilon_{2,1/2}, \boxed{7} \omega_{21}, \boxed{8} z_{2,1/2}, \boxed{22} e_{21}, \boxed{23} \varepsilon_{2,3/2}, \\
& \boxed{26} z_{2,3/2}
\end{aligned}$$

Продолжение рис.4

#### СПИСОК ЛИТЕРАТУРЫ

1. Верлань А.Ф., Сизиков В.С. Интегральные уравнения: методы, алгоритмы, программы. Киев, Наукова думка, 1986.
2. Валях Е. Последовательно-параллельные вычисления, М.: Мир, 1985, 456 с.
3. Седухин С.Г. Параллельная интерпретация прямых методов линейной алгебры. Программирование, 1984, № 4, с.57-58.
4. Нерсесян А.Б. Новые алгоритмы численного решения интегральных уравнений второго рода. ДАН, Арм.ССР, 1989, т.89, № 4.

Рукопись поступила 8 января 1990 г.

М. А. МХИТАРЯН, А. Б. НЕРСЕСЯН

ПАРАЛЛЕЛЬНО-ПОТОЧНАЯ ИНТЕРПРЕТАЦИЯ НЕКОТОРЫХ АЛГОРИТМОВ  
ТРЕУГОЛЬНОГО РАЗЛОЖЕНИЯ ИНТЕГРАЛЬНЫХ ОПЕРАТОРОВ ВТОРОГО РОДА

Редактор Л. П. Мукаян

Технический редактор А. С. Абрамян

---

Подписано в печать 29/III-90г.	ВФ-01347	Формат 60x84/16
Офсетная печать. Уч. изд. л. 0,8	Тираж 299 экз.	Ц. 10 к.
Зак. тип. № 175	Индекс 3649	

---

Отпечатано в Ереванском физическом институте  
Ереван 36, ул. Братьев Алиханян, 2

The address for requests:  
Information Department  
Yerevan Physics Institute  
Alikhanian Brothers 2,  
Yevan, 375036  
Armenia, USSR

