

УДК 681.325

ПОСЛЕДОВАТЕЛЬНАЯ И ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ
МЕТОДА ВЕРХНЕЙ РЕЛАКСАЦИИ ПРИ РЕШЕНИИ
ЗАДАЧ МАТЕМАТИЧЕСКОЙ ФИЗИКИ

Каневский Ю.С., Лепеха В.Л., Сергиенко А.М. (Национальный технический университет Украины “КПИ”, Киев,

Выжиковски Р. (Ченстоховский политехнический ин-т, Польша)

Рассмотрена реализация метода верхней релаксации при решении задач математической физики методом конечных элементов. Проанализированы возможные варианты записи матриц в разреженном формате. Показаны последовательная и параллельная формы реализации данного метода. Приведены экспериментальные результаты решения задач математической физики с помощью данного метода.

Анотація

Розглянуто реалізацію методу верхньої релаксації при розв’язуванні задач математичної фізики методом кінцевих елементів. Проаналізовано можливі варіанти запису матриць в розрідженому форматі. Показано послідовну та паралельну форми реалізації даного методу. Наведено експериментальні результати розв’язування задач математичної фізики з використанням даного методу.

Ключевые слова: метод конечных элементов, разреженная матрица, метод верхней релаксации.

Введение. Решение крупных научно-технических проблем стало возможным благодаря применению математического моделирования и новых численных методов. Быстрое развитие микроэлектроники и создание высокопроизводительных ЭВМ привели к появлению нового способа теоретического исследования сложных процессов, допускающих математическое описание, - вычислительного эксперимента [1].

При решении задач математической физики (моделировании тепловых, электромагнитных, механических и др. процессов) достаточно часто используется метод конечных элементов (МКЭ). При всем своем разнообразии задача, в конечном итоге, сводится к решению (зачастую многократному) систем линейных алгебраических уравнений (СЛАУ) ($Ax = y$), где A – является сильно разреженной матрицей.

С целью экономии памяти и ускорения вычислительного процесса целесообразно оперировать матрицами, записанными в разреженных форматах. При записи матрицы в разреженном формате для решения системы линейных алгебраических уравнений (СЛАУ) целесообразно использовать итерационные методы, поскольку применение прямых методов влечет за собой изменение структуры матрицы, и, как правило, приводит матрицу к ленточному или плотно заполненному виду [2].

Задачи математической физики предполагают большой объем вычислений. При этом сложность современных задач, выраженная в числе математических операций, и требования к точности полученных результатов (что при использовании итерационных методов решения СЛАУ также увеличивает необходимый объем вычислений) растут быстрее, чем производительность вычислительных средств. Поэтому, учитывая все возрастающий объем вычислений и фундаментальные ограничения на скорость переключения элементов и распространения сигналов, достаточно актуальной проблемой становится задача распараллеливания вычислений.

Данная статья посвящена последовательной и параллельной реализации метода верхней релаксации.

Краткое описание метода. Метод верхней релаксации относится к треугольным методам [1]. В нем оператору В соответствует нижняя треугольная матрица $B = D + \omega L$, где D - диагональная матрица; L - нижняя треугольная матрица из разложения исходной матрицы $A = L+D+U$; ω - коэффициент релаксации (в случае верхней релаксации $1 < \omega < 2$).

Рассматриваемый метод хорошо известен, поэтому при его описании ограничимся результирующей вычислительной формулой [1]:

$$a_{ii} \bullet x_i^{(k+1)} = (1 - \omega) \bullet a_{ii} \bullet x_i^{(k)} - \omega \bullet \left(\sum_{j=1}^{i-1} a_{ij} \bullet x_j^{(k+1)} + \sum_{j=i+1}^M a_{ij} \bullet x_j^{(k)} \right) + \omega \bullet y_i, \quad (1)$$

где ω - коэффициент релаксации; k - номер итерации; M - размер матрицы; i, j - 1, 2, ..., M ; найденное $x_i^{(k+1)}$ сохраняется с замещением $x_i^{(k)}$.

Поскольку на матрицу не накладывается требование симметричности, ее целесообразно нормализовать. Тогда результирующая формула примет следующий вид:

$$x_i'^{(k+1)} = (1 - \omega) \bullet x_i'^{(k)} - \omega \bullet \left(\sum_{j=1}^{i-1} a_{ij}' \bullet x_j'^{(k+1)} + \sum_{j=i+1}^M a_{ij}' \bullet x_j'^{(k)} \right) + \omega \bullet y_i', \quad (2)$$

где $a_{ij}' = a_{ij}/a_{ii}$, $y_i' = y_i/a_{ii}$.

Метод верхней релаксации отличается простотой реализации и экономичностью. К его недостаткам следует отнести определенную сложность при выборе оптимального коэффициента релаксации.

Выбор формата представления матрицы. Под разреженным форматом представления подразумевают такую запись матрицы, при которой в памяти ЭВМ хранятся только значения ненулевых элементов и их место расположения в матрице. В настоящее время описано достаточно много таких форматов [3]. Однако специфика реальных задач или матриц может потребовать конкретного представления матрицы. Опишем подробно представление матрицы в разреженном строчном формате RR(C(O)), который можно достаточно эффективно применять в методе конечных элементов.

В данном формате значения ненулевых элементов и соответствующие им столбцовые индексы хранятся по строкам в двух массивах, например, для матрицы X соответственно в массивах XN и JX . Кроме того, в $RR(C(O))$ используется массив указателей (IX), i -й элемент которого указывает позицию в предыдущих двух массивах, с которой начинается описание i -й строки матрицы X . В этом формате произвольная матрица

$$X = \begin{pmatrix} x_{11} & 0 & 0 & x_{14} & 0 \\ 0 & x_{22} & 0 & x_{24} & x_{25} \\ 0 & x_{32} & x_{33} & 0 & 0 \\ x_{41} & 0 & 0 & x_{44} & 0 \\ 0 & 0 & 0 & 0 & x_{55} \end{pmatrix}$$

будет представлена в виде трех массивов:

$$\begin{array}{l} IX: \quad 1 \quad 3 \quad 6 \quad 8 \quad 10 \quad 11 \\ JX: \quad 1 \quad 4 \quad 2 \quad 4 \quad 5 \quad 2 \quad 3 \quad 1 \quad 4 \quad 5 \\ XN: \quad x_{11} \quad x_{14} \quad x_{22} \quad x_{24} \quad x_{25} \quad x_{32} \quad x_{33} \quad x_{41} \quad x_{44} \quad x_{55} \end{array} \quad (3)$$

Описание реализующей программы. При решении СЛАУ матрица предварительно нормализуется и представляется в виде (3). Правая часть, разделенная на диагональ, записана в векторе y . Искомое решение записывается в векторе x . Вычисления будем производить по формуле (2), которая будет представлена в виде

$$x_i = x_i + \omega \left(y_i - \sum_{j=1}^N a_{ij} x_j \right), \quad (4)$$

причем вновь вычисленное значение x_i будет записываться на место старого.

Рассмотрим фрагмент (основной цикл) программы, написанной на языке Си. В нем $e2$ - заданная погрешность вычислений, n - число строк матрицы A ; w - коэффициент релаксации; e, i, j, s - промежуточные параметры.

```

while(e > e2)
{
    /* вычисление  $X^{(k+1)}$  */
    for (i = 1; i < n; i++)
    {
        /* вычисление  $X_i^{(k+1)}$  */
        s = Y[i] - X[i];
        for (j = IA[i]; j < IA[i+1]; j++)
            s = s - AE[AN[j]] * X[JA[j]];
        X[i] = X[i] + w * s;
    }
    /* вычисление e - проверка на сходимость */
    void shod(); shod();
}

```

где *shod()* - процедура проверки на сходимость, например, $e = \text{MAX}(Ax-y)$.

Элементы вектора X вычисляют, начиная с первого. Первоначально параметру s присваивают значение $Y[i] - X[i]$, поскольку элемент $a_{ii} = 1$. Затем вычисляют значение, записанное в скобках формулы (4). Для этого параметру j присваивают адрес первого элемента i -й строки. Все элементы строки расположены последовательно, начиная с $IA[i]$ и заканчивая $IA[i+1]-1$. На j позиции векторов JA и AN находятся адреса соответственно элементов векторов X и AE , которые должны быть перемножены между собой и накоплены в s . По окончании цикла накопления вычисленное значение умножается на коэффициент релаксации w и суммируется со значением $X[i]$. По окончании вычисления вектора $X^{(k+1)}$ проводится оценка сходимости в соответствии с выбранным критерием. Объем вычислений, необходимый для такой оценки, может оказаться не ниже вычислительной сложности формирования вектора X , поэтому в некоторых случаях проводить оценку целесообразно через определенное число итераций.

Рассмотрим двумерную задачу упругости с двумя степенями свободы (изгиб балки под воздействием сосредоточенной нагрузки (рис.1)). Проанализируем два

варианта: с предварительным оценочным расчетом $X^0 = F(x, y, P)$ и при $X^0 = 0$. Полученные расчетные данные сведены в табл. 1 (здесь x, y - параметры, определяющие сетку; в ячейках указано число итераций, необходимых для решения задачи при $\varepsilon = \text{MAX}(Ax - y) = 0,0001$).

Таблица 1

x	30	50	80	40	80
y	100	100	100	200	200
$X^0 = F(x, y, P)$	111	108	112	203	185
$X^0 = 0$	581	852	438	2201	1202

Как видно из табл. 1 метод верхней релаксации сходится очень быстро при наличии приближенного решения. В противном случае, число итераций может быть достаточно большим.

Распараллеливание вычислений. Решетчатый граф данного алгоритма размещается в параллелепипеде размерности $N \times M \times K$.

Граф алгоритма выполнения k -й итерации не намного отличается от графа умножения матрицы на вектор. Как видно из рис. 2,а, граф является регулярным. Базовая операция, соответствующая каждому узлу графа, имеет вид: $S = S + a \bullet b$. Учитывая разреженность матрицы, спроецируем граф по оси i , с тем чтобы в каждом узле находилась строка матрицы A и соответствующий ей элемент вектора y (рис 2,б). Такая схема решения задачи имеет существенный недостаток. Как известно, в методе верхней релаксации при вычислении элемента x_i используются уже вычисленные в этой итерации элементы $x_j (j < i)$. В связи с этим, при переходе к вычислению следующего элемента необходимо вводить задержку (на $N-1$ тактов) для ожидания вновь вычисленного элемента.

Частично трансформируем исходный граф (рис. 3,а). Спроецируем его вновь на ось i (рис. 3,б.). В данной ситуации задержка в каждом узле составит 1 такт. Такая схема вычислений будет достаточно эффективной при плотно заполненной матрице. В случае применения линейной вычислительной структуры (рис. 4), соответствующей графу, приведенному на рис. 3,б, загрузка процессорных элементов (ПЭ) составит - $N/(N+1)$. Однако для разреженной

матрицы, загруженность i -го ПЭ будет равна - $r_i/(N+1)$, где r_i - число ненулевых элементов в i -й строке. Учитывая, что $r_i \ll N$, можно сделать вывод о том, что выполнение данной задачи на структуре с N ПЭ крайне не эффективно.

Значительно больший интерес может вызвать ограниченный параллелизм (решение задачи на K ($K \ll N$) ПЭ). Для этого склеим соседние вершины графа таким образом, чтобы в графе, приведенном на рис. 3,б осталось K вершин. Структурная схема такого вычислителя аналогична рассмотренной (см. рис. 4), с той лишь разницей, что число ПЭ равно K . В памяти процессорного элемента находятся s ($s = N/K$) строк матрицы A , s элементов вектора y , а также те элементы вектора x , которые участвуют в вычислениях, проводимых в данном ПЭ. По мере вычисления значений вектора x , они передаются в другие ПЭ. В данном случае загруженность ПЭ будет не меньше $\sum r_i / (\sum r_i + N)$. При этом желательно склеивать вершины графа таким образом, чтобы число передач между ПЭ было минимальным. Процессорные элементы должны работать в синхронном режиме (каждый ПЭ начинает $(k+1)$ -ю итерацию только после получения всех значений, вычисленных в других ПЭ во время k -й итерации).

В табл. 2 показано расписание работы вычислительной структуры, состоящей из трех ПЭ, при решении СЛАУ с матрицей, изображенной на рис. 5. В 1-й колонке показан номер такта работы структуры, во 2-, 4-, 6-й – операция выполняемая в ПЭ, в 3-, 5-й – передаваемое значение; стрелочкой указано направление передачи. Предполагается, что вновь вычисленное значение x_i передается в ПЭ, где оно будет использовано сразу же после и на фоне вычислений. Такую возможность обеспечивают, например, процессорные матрицы на базе сигнальных процессоров типа TMS320C40, DSP9600, ADSP21000 и др. Как видно, в случае линейной структуры с локальными связями и при возможности обмена информацией на фоне вычислений можно добиться практически 100% загрузки оборудования.

Если обмен информацией на фоне вычислений невозможен или при построении параллельной вычислительной системы с общей шиной то загруженность ПЭ можно определить как $K_g / (K_g + K_n)$, где K_g - число вычислений в ПЭ, K_n - число необходимых передач после каждой итерации.

Оценим загруженность ПЭ на примере решения описанной задачи упругости при $x = 80$, $y = 200$ (с учетом двух степеней свободы - решающее поле - 160×200). При распараллеливании вычислений и загрузке ПЭ, как это видно из рис. 4, каждый ПЭ (кроме крайних) по окончании итерации должен передать соседним 320 вновь вычисленных значений и принять также 320 значений, вычисленных в соседних ПЭ. Таким образом, принимают 160 значений. Рассмотрим загруженность линейной структуры с локальными связями и загруженность структуры с общей шиной. При этом будем предполагать, что время, необходимое для выполнения операции пересылки, равно времени выполнения арифметической операции. Загруженность арифметико-логического устройства (АЛУ) процессорных элементов условно не учитывается.

Рассмотрим распараллеливание задачи на четырех ПЭ. В каждом ПЭ вычисляются 8000 значений вектора x , при этом для вычисления каждого элемента вектора необходимо выполнить 15 арифметических операций. Тогда число арифметических операций, выполняемое в каждом ПЭ для одной итерации, $K_B = 120000$. При этом число операций пересылки для линейной структуры - $K_P = 640$, а для структуры с общей шиной - $K_P = 960$. Загруженность в данном случае практически стопроцентная. Аналогично можно получить K_B и K_P для другого числа ПЭ.

В табл. 3, приведены соответствующие данные при отображении данной задачи на различное число ПЭ.

Таблица 3

Число ПЭ	Вычислительная структура					
	Линейная			С общей шиной		
	K_B	K_P	загруженность ПЭ	K_B	K_P	загруженность ПЭ
4	120000	640	$\approx 100\%$	120000	960	$\approx 99\%$
8	60000	640	$\approx 99\%$	60000	2240	$\approx 96\%$
16	30000	640	$\approx 98\%$	30000	4800	$\approx 86\%$
32	15000	640	$\approx 96\%$	15000	9920	$\approx 60\%$

Как видно из таблицы 3, загруженность вычислительной структуры с локальными связями достаточно высока даже при большом числе ПЭ. Загруженность же структуры с ОШ резко снижается при увеличении числа ПЭ, а с учетом времени для организации обмена сообщениями (на практике время, необходимое для пересылки одного значения, равно времени выполнения 10 - 1000 арифметических действий) она может упасть настолько, что возникает вопрос о целесообразности самого распараллеливания.

При существенном влиянии пересылок на снижение загруженности ПЭ интерес вызывает асинхронный режим вычислений. В таком режиме межпроцессорный обмен информацией выполняется после выполнения несколько итераций (склеивание итераций). Такой режим работы при построении вычислительной структуры с локальными связями и фоновым обменом информацией исключил бы необходимость синхронизации двух процессов (вычисления и передачи информации), что упростило бы программирование задачи. При построении структуры с ОШ уменьшение операций пересылки привело бы к повышению загруженности вычислительной структуры. Метод верхней релаксации предполагает возможность блочной реализации, и, поэтому, при таком склеивании задача будет сходиться [2]. В табл. 4 приведено число итераций, необходимых для решения описанной задачи при различном числе ПЭ и склеенных итераций.

Таблица 4

ПЭ	Число склеенных итераций			
	1	2	3	4
8	185	508	672	856
16	185	678	891	1644
32	185	880	1026	1740

Как видно из табл. 4, при использовании метода верхней релаксации строить асинхронно работающую параллельную вычислительную систему нецелесообразно, поскольку резко возрастает число необходимых для решения СЛАУ итераций, и, как следствие – число обменов информацией.

Заключение. Для решения задач математической физики с записью матриц в разреженном строчном формате для решения СЛАУ можно применять метод верхней релаксации. Метод достаточно прост в реализации на последовательных вычислительных средствах и не требует избыточной информации. Он особенно эффективен при наличии приближенного решения, например, при решении нелинейных задач, предполагающих последовательное решение большого числа СЛАУ. При этом решение каждой системы является приближительным решением последующей. Метод хорошо распараллеливается. В случае применения линейной вычислительной структуры с локальными связями достигается практически 100%-я загруженность. При использовании структуры с общей шиной ее загруженность зависит от объема и скорости передачи информации и может быть низкой.

Литература

1. Самарский А.А., Николаев Е.С., Методы решения сеточных уравнений. - М.: Наука, 1978. - 592 с.
2. Воеводин В.В., Кузнецов Ю.А., Матрицы и вычисления. – М.: Наука, 1984. - 320 с.
3. Писанецки С. Технология разреженных матриц. - М.: Мир, 1988. – 411 с.

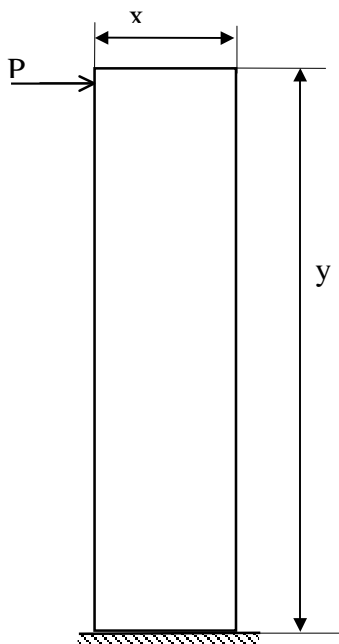


Рис. 1. Схема демонстрационной задачи

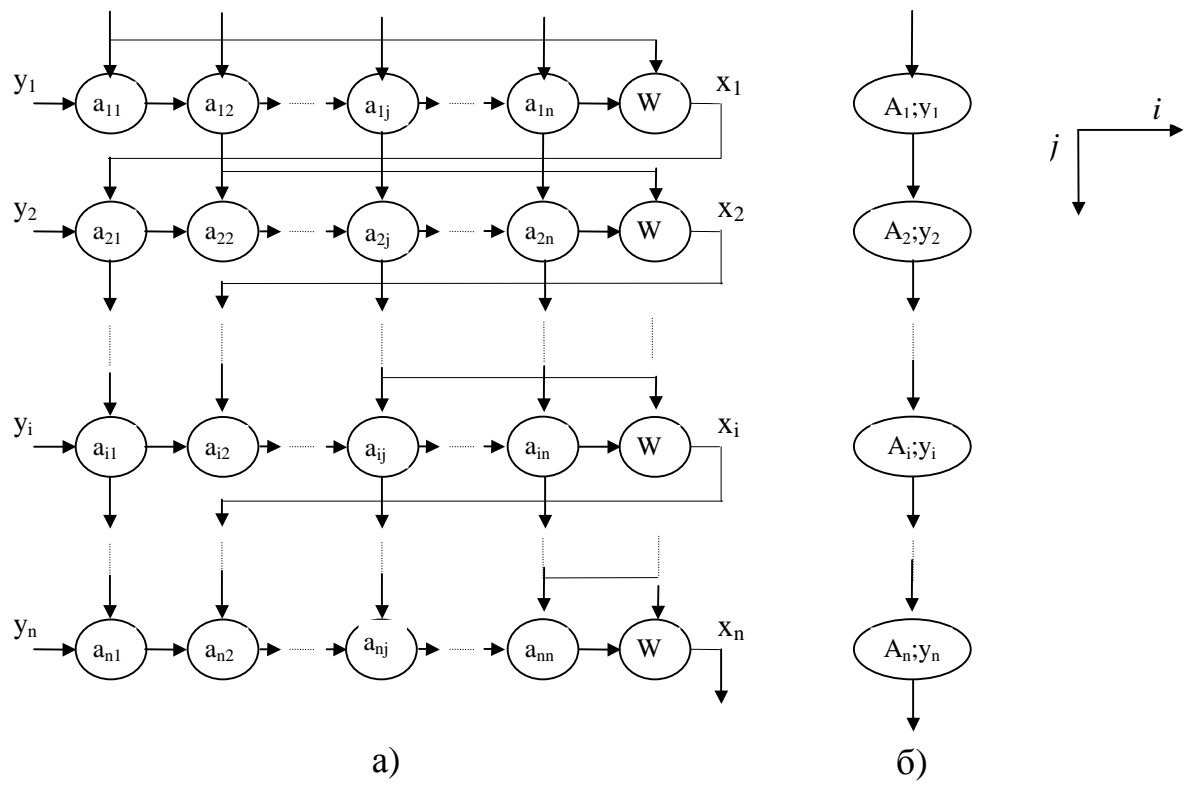


Рис. 2. Граф k -й итерации (а) и его проекция по оси i (б).

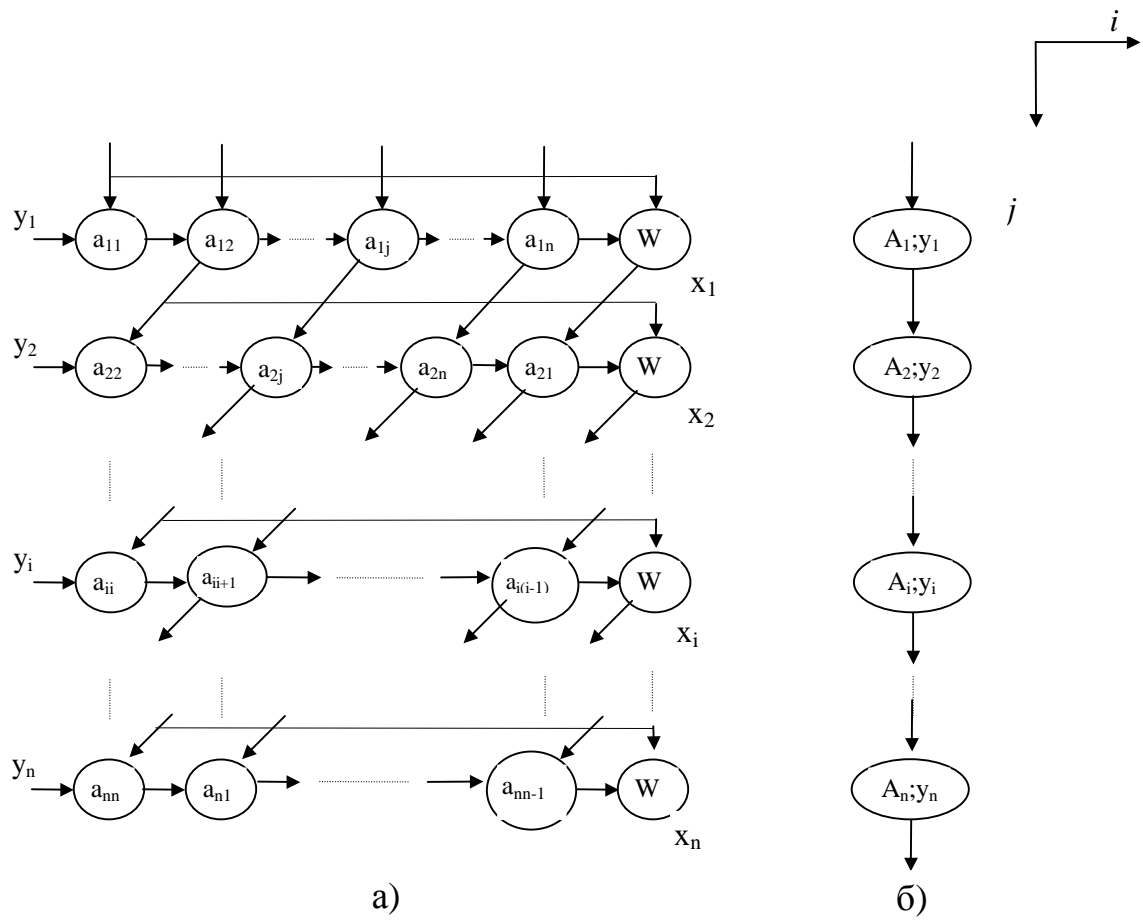


Рис. 3. Граф k -й итерации (а) и его проекция по оси i (б).

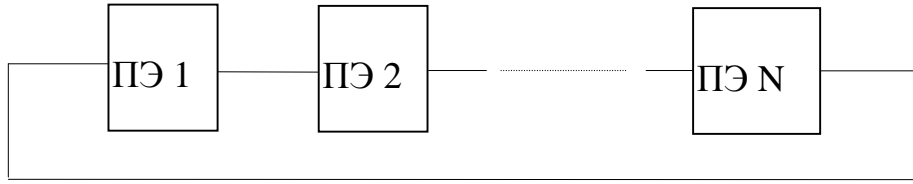


Рис. 4 Структурная схема вычислителя

	1	2	3	4	5	6	7	8	9	10	11	12	
1	x	x		x									ПЭ1
2	x	x	x		x								
3		x	x	x		x							
4	x		x	x	x		x						
5		x		x	x	x		x					ПЭ2
6			x		x	x	x		x				
7				x		x	x	x		x			
8					x		x	x	x		x		
9						x		x	x	x		x	ПЭ3
10							x		x	x	x		
11								x		x	x	x	
12									x		x	x	

Рис. 5. Портрет матрицы и разбиение ее по ПЭ

Таблица 2

	ПЭ1		ПЭ2		ПЭ3
1	$s = y_1 - x_1$				
2	$s = s - x_2 \bullet a_{12}$				
3	$s = s - x_4 \bullet a_{14}$				
4	$x_1 = x_1 + w \bullet s$				
5	$s = y_2 - x_2$	$x_1 \rightarrow$			
9	$x_2 = x_2 + w \bullet s$				
10	$s = y_3 - x_3$	$x_2 \rightarrow$			
15	$s = y_4 - x_4$	$x_3 \rightarrow$			
20	$x_4 = x_4 + w \bullet s$		$s = y_5 - x_5$		
21		$x_4 \rightarrow$	$s = s - x_2 \bullet a_{52}$		
25	$s = y_1 - x_1$		$x_5 = x_5 + w \bullet s$		
26	$s = s - x_2 \bullet a_{12}$	$\leftarrow x_5$	$s = y_6 - x_6$		
29	$s = y_2 - x_2$	$x_1 \rightarrow$	$s = s - x_7 \bullet a_{67}$		
32	$s = s - x_5 \bullet a_{25}$	$\leftarrow x_6$	$s = y_7 - x_7$	$x_6 \rightarrow$	
33	$x_2 = x_2 + w \bullet s$		$s = s - x_4 \bullet a_{74}$		
34	$s = y_3 - x_3$	$x_2 \rightarrow$	$s = s - x_6 \bullet a_{76}$		
38	$x_3 = x_3 + w \bullet s$	$\leftarrow x_7$	$s = y_8 - x_8$	$x_7 \rightarrow$	
39	$s = y_4 - x_4$	$x_3 \rightarrow$	$s = s - x_5 \bullet a_{85}$		
43	$s = s - x_7 \bullet a_{47}$		$x_8 = x_8 + w \bullet s$		$s = y_9 - x_9$
44	$x_4 = x_4 + w \bullet s$	$\leftarrow x_8$	$s = y_5 - x_5$	$x_8 \rightarrow$	$s = s - x_6 \bullet a_{96}$
45		$x_4 \rightarrow$	$s = s - x_2 \bullet a_{52}$		$s = s - x_8 \bullet a_{98}$
49	$s = y_1 - x_1$		$x_5 = x_5 + w \bullet s$	$\leftarrow x_9$	$s = y_{10} - x_{10}$
50	$s = s - x_2 \bullet a_{12}$	$\leftarrow x_5$	$s = y_6 - x_6$		$s = s - x_7 \bullet a_{(10)7}$
53	$s = y_2 - x_2$	$x_1 \rightarrow$	$s = s - x_7 \bullet a_{67}$		$x_{10} = x_{10} + w \bullet s$
54	$s = s - x_1 \bullet a_{21}$		$s = s - x_9 \bullet a_{69}$	$\leftarrow x_{10}$	$s = y_{11} - x_{11}$
55	$s = s - x_3 \bullet a_{23}$		$x_6 = x_6 + w \bullet s$		$s = s - x_8 \bullet a_{(11)8}$
56	$s = s - x_5 \bullet a_{25}$	$\leftarrow x_6$	$s = y_7 - x_7$	$x_6 \rightarrow$	$s = s - x_{10} \bullet a_{(11)(10)}$
57	$x_2 = x_2 + w \bullet s$		$s = s - x_4 \bullet a_{74}$		$s = s - x_{12} \bullet a_{(11)(12)}$
58	$s = y_3 - x_3$	$x_2 \rightarrow$	$s = s - x_6 \bullet a_{76}$		$x_{11} = x_{11} + w \bullet s$
59	$s = s - x_2 \bullet a_{32}$		$s = s - x_8 \bullet a_{78}$	$\leftarrow x_{11}$	$s = y_{12} - x_{12}$
62	$x_3 = x_3 + w \bullet s$	$\leftarrow x_7$	$s = y_8 - x_8$	$x_7 \rightarrow$	$x_{12} = x_{12} + w \bullet s$