

УДК 518.5:681.142.001.2

УНИВЕРСАЛЬНАЯ ПРОГРАММА ОПТИМИЗАЦИИ ЭЛЕКТРИЧЕСКИХ ЦЕПЕЙ

В.К. Королев

Описывается программа, позволяющая решать задачи оптимизации параметров как статических, так и динамических режимов электрических цепей, причем целевая функция может быть как гладкой, так и негладкой; на оптимизируемые параметры накладываются двусторонние ограничения. Программа написана на языке АКИ.

1. Введение

В работе [1] приведена универсальная программа оптимизации параметров динамических систем квадратичным методом. Программа названа универсальной в том смысле, что она осуществляет решение описанной в [1] задачи стандартным способом; нестандартным в программе является блок счета правых частей системы уравнений и блок расчета начального значения подынтегральной функции.

Предлагаемая ниже программа универсальна в более широком смысле. Во-первых, рассматривается довольно общий вид функционала, играющего роль критерия качества. Во-вторых, от целевой функции не требуется непрерывной дифференцируемости. В-третьих,

программа может быть применена для оптимизации динамических и статических задач как во временной, так и частотной областях.

В тех задачах, где вычисление целевой функции занимает много машинного времени, остро встает вопрос о минимальном числе вычислений этой функции и максимальном использовании информации, полученной от предыдущих вычислений. Способ решения задачи оптимизации, реализованный в описываемой программе, отвечает этому требованию.

2. Постановка задачи

Оптимизация параметров электрической цепи — один из этапов проектирования, когда известна ее конфигурация и проведено математическое моделирование.

В общем случае мы имеем дело со следующей системой уравнений, описывающих электрическую цепь:

$$\begin{cases} \dot{x} = \varphi(x, y, z, t), \\ \frac{dy}{dt} = f(x, y, z, t), \quad y(t_0) = y^0, \end{cases} \quad (1)$$

где t — скаляр (время), x , y и z — векторы, φ и f — векторные функции. Физический смысл x , y и z : $x^T = \|x_1, \dots, x_n\|^T$ — вектор параметров электрической цепи (сопротивления, емкости и т.п.), $y^T = \|y_1, \dots, y_n\|^T$ и $z^T = \|z_1, \dots, z_n\|^T$ — векторы состояний цепи (токи, напряжения и т.д.) (знак "T" означает транспонирование).

Занись уравнений (1) в виде системы конечных и дифференциальных уравнений отражает как формальную, так и существенную стороны дела. Если отсутствуют "дифференциальные" переменные y и соответствующие им уравнения, то мы имеем описание статического режима цепи: $\dot{x} = \varphi(x, z)$. С другой стороны, не всегда возможно дифференциальные уравнения электрической цепи привести к нормальному виду: $\frac{dy}{dt} = f(x, y, t)$ (как, например, в [2]). Так, если в [3-4] запись уравнений цепи в виде (1) диктовалась соображениями упрощения выкладок при получении уравнений в вариациях, в [1] существенной необходимости оставлять

"конечные" переменные x не было, то в [2] исключить x явно из системы (I) было невозможно.

Таким образом, мы приходим к выводу, что система (I) представляет наиболее общий вид уравнений электрической цепи, учитывающий как статический, так и динамический режимы.

Система (I) рассматривается на множестве $X \times Y \times Z \times T$, где

$$X = \{x | \alpha \leq x \leq \beta\}, \quad (2)$$

m - мерный параллелепипед;

$Y \times Z$ - ограниченная связная область в евклидовом пространстве E^{n+s} ;

$T = \{t | t_0 \leq t \leq t_N\}$ - конечный интервал.

Предполагается, что якобиан $J = \frac{\partial(\varphi - z_1, \dots, \varphi - z_s)}{\partial(x_1, \dots, x_s)}$ отличен от нуля в рассматриваемой области задания системы (I). Это обеспечивает [5] в каждой точке области существование однозначной функции $x = x(x, y, t)$, которая может быть определена, вообще говоря, только численно.

Далее, предположим, что правые части системы (I) непрерывны и непрерывно дифференцируемы по y и x в области ее задания. Тогда существует единственное решение задачи Коши для (I) [6]. Мы не будем предполагать, что правые части системы (I) непрерывно дифференцируемы по параметрам, так что это решение может и не быть непрерывно дифференцируемым по параметрам.

Кроме описания электрической цепи во временной области системой дифференциальных уравнений, в теории линейных цепей широко применяется описание цепи в частотной области заданием частотной характеристики $\Phi = \Phi(\omega)$. Параметры входят при этом в выражение Φ .

При проектировании цепи ставится задача так подобрать ее параметры, чтобы выходная функция (в частотной области - частотная характеристика) аппроксимировала некоторую заданную функцию. За меру близости двух функций $y(t)$ и $y_*(t)$ обычно принимается расстояние $\rho(y, y_*)$ либо в метрике функционального пространства L^2 :

$$\rho^2(y, y_*) = \int_{t_0}^{t_N} [y(t) - y_*(t)]^2 dt, \quad (3)$$

либо в чебышевской метрике для пространства непрерывных на отрезке функций:

$$\rho(y, y_*) = \max_{t \in [t_0, t_N]} |y(t) - y_*(t)|. \quad (4)$$

Такого же типа функционалы могут быть использованы для определения расстояний между $\Phi(\omega)$ и $\Phi_*(\omega)$ в частотной области [7].

(3) есть частный случай функционала более общего типа:

$$F = \int_{t_0}^{t_N} f(x, y, z, t) dt, \quad (5.1)$$

(который мы назовем функционалом типа Лагранжа [8]). Наряду с (5.1) рассмотрим еще два [8]:

$$F = y_1(t_N) \quad (5.2)$$

(функционал типа Майера) и

$$F = \int_{t_0}^{t_N} f_0(x, y, z, t) dt + g(t_0, y(t_0), z(t_0); t_N, y(t_N), z(t_N))$$

(функционал типа Больца).

Задачи с (3) мы уже рассматривали [1-4]. К задачам с функционалом типа (5.2) относятся, например, задачи на быстрое действие: $F = t_N - t_0$ (см. ниже - пример 2). Функционал III - го типа наиболее общий, охватывающий первые два. На самом деле все три обладают одинаковой степенью общности и переводятся один в другой заменой переменных [8].

На решении системы (I) функционал F становится функцией параметров $F = F(x)$, которая, вообще говоря, может и не быть непрерывно дифференцируемой (гладкой) функцией своих ар-

гументов по двум причинам: либо из-за недифференцируемости по параметрам решения системы (1), либо недифференцируемости функционала (например, типа (4)).

В задаче оптимизации параметров требуется найти минимум целевой функции $F(x)$ в области (2):

$$\min \{F(x) | x \in \mathcal{X}\} \quad (6)$$

Таким образом, получаем задачу нелинейного математического программирования, в которой допустимая область (2) выпукла и весьма проста, а целевая функция задана довольно сложным способом: как функционал на решении системы дифференциальных уравнений (для статических или задач в частотной области, вообще говоря, указывается алгоритм вычисления целевой функции).

В общем случае мы ничего не можем сказать о выпуклости функции $F(x)$, значения которой получаются только численно. Поэтому задача (6) ставится лишь на отыскание локального минимума. Некоторые сведения о характере множества минимумов функции $F(x)$ в области (2) можно получить, беря различные исходные допустимые точки.

3. Алгоритм решения задачи

Каким применяемого ниже метода решения задачи оптимизации является алгоритм РЗ Зойтендейка [9], идея которого заключается в учете дополнительных ограничений — условий сопряженности, улучшающих (по Зойтендейку) сходимость в общем случае нелинейной функции цели. В случае квадратичной функции с неотрицательно определенной матрицей имеет место конечная сходимость.

В нашей литературе известен аналогичный метод сопряженных градиентов (см. [10]), где он применяется для решения систем линейных уравнений).

В американской литературе [11] приводится подобная процедура, дающая конечную сходимость для квадратичной функции. Отметим, что в [11] решается задача без ограничений.

Мы несколько упростим алгоритм Зойтендейка, поскольку ог-

раничения в нашем случае очень просты. Мы не будем заниматься доказательством сходимости предлагаемого процесса, представляющего эвристическую процедуру, на основе которой составлена довольно обшая программа для решения широкого круга практических задач оптимизации.

Во всяком итерационном процессе движения к оптимуму различаются три этапа: 1) нахождение в текущей допустимой точке направления движения, 2) определение величины шага по этому направлению, 3) проверка критерия оптимальности.

Опишем, как осуществляются эти этапы в нашем случае, а потом приведем весь алгоритм в целом. Для простоты начнем с 3-го этапа.

Процесс достижения минимума считается законченным, если мы пришли в такую его окрестность, что относительное уменьшение целевой функции $\delta F = \frac{F^k - F^{k+1}}{1/2(F^{k+1} + F^k)}$ становится меньше наперед заданного числа ε . Кроме этого, используется еще и другой критерий оптимальности: малость модуля текущего направления S^k :

$$\|S^k\|^2 = \sum_i (S_i^k)^2 < \varepsilon'$$

Предположим, что в текущей допустимой точке x^k известно возможное подходящее направление S^k (терминология из [9]). Шаг λ_k по направлению S^k определяется так:

$$\lambda_k = \min(\lambda_k', \lambda_k''),$$

где λ_k' — то значение λ , при котором луч $x^k + \lambda S^k$ "протыкает" границу допустимой области:

$$\lambda_k' = \min_i \lambda_{ki}, \quad \lambda_{ki} = \begin{cases} \frac{a_i - x_i^k}{S_i^k}, & \text{если } S_i^k < 0, \\ \frac{b_i - x_i^k}{S_i^k}, & \text{если } S_i^k > 0; \end{cases}$$

λ''_k определяется из условия, что функция $f(\lambda) \equiv F(x^k + \lambda S^k)$ при $\lambda = \lambda''_k$ достигает минимума на отрезке $[0, \lambda'_k]$.

Для определения λ''_k $f(\lambda)$ аппроксимируется параболой $f(0) + f'(0)\lambda + \frac{1}{2}\lambda^2$. Величины $f(0) = F(x^k)$ и $f'(0) = \nabla f(x^k)^T S^k$ при этом уже вычислены (см. ниже), а для определения коэффициента δ вычисляется значение $f(\lambda'_k)$. Тогда:

$$\delta = \frac{f(\lambda'_k) - f(0) - \nabla f(x^k)^T S^k \lambda'_k}{\lambda'^2_k} \quad \text{и} \quad \lambda''_k = -\frac{\nabla f(x^k)^T S^k}{\delta}.$$

Подробности этой процедуры приведены в блок-схеме (рис. 8) и автокодовой программе.

Для определения направления S^k вычисляется градиент целевой функции: $g^k = \text{grad } F(x) |_{x^k}$. Если $\lambda_{k-1} = \lambda'_{k-1}$, т.е. мы вышли на граничную гиперплоскость по i^* -й координате, то полагаем $g_{i^*}^k = 0$, если направление $-g_{i^*}^k$ выводит за пределы области (2). Вектор возможного подходящего направления при $\lambda_{k-1} = \lambda'_{k-1}$ есть антиградиент: $S_i^k = -g_i^k$, $i = 1, \dots, m$ (в начале процесса полагаем $\lambda_1 = \lambda'_1$).

Если $\lambda_{k-1} = \lambda''_{k-1} < \lambda'_{k-1}$, то мы учитываем дополнительные ограничения - условия сопряженности:

$$(g^{\ell+1} + S^{\ell+1})^T S = 0, \quad (7)$$

$\ell = 2, \dots, k-1$, где ℓ - таково, что $\lambda_{\ell-1} = \lambda'_{\ell-1}$, $\lambda_\ell = \lambda''_\ell$, $\ell \leq k \leq k-1$.

Запишем (7) в матричной форме:

$$AS = 0. \quad (8)$$

На каждом шаге итерационного процесса мы сохраняем столько условий сопряженности (7), чтобы оставалась, по крайней мере, одна степень свободы при выборе S^k , т.е. матрица A имеет размерность $p \times m$; $p \leq m-1$.

Пусть A_2 - невырожденная подматрица матрицы A размерности $p \times p$. Представим A в виде:

$$A = \|A_1, A_2, A_3\|, \quad (9)$$

где

$$A_1 = \|\alpha_1, \dots, \alpha_{p_1-1}\|,$$

$$A_2 = \|\alpha_{p_1}, \dots, \alpha_{p_1+(p-1)}\|,$$

$$A_3 = \|\alpha_{p_1+p}, \dots, \alpha_m\|.$$

(точка означает, что соответствующий индекс пробегает все значения).

В соответствии с представлением (9) векторы g^k и S также разобьем на части:

$$g^k = \|g_1^k, g_2^k, g_3^k\|; \quad S = \|S_1, S_2, S_3\|,$$

(заметьте, что если невырожденная подматрица A_2 стоит в начале или в конце матрицы A , то мы имеем разбиение A , g^k и S на две части).

Условия сопряженности (8) принимает вид:

$$A_1 S_1 + A_2 S_2 + A_3 S_3 = 0.$$

Оставшаяся за свободными (внебазисными) переменными S_1^k и S_3^k их значения, найденные при вычислении градиента:

$$S_1^k = -g_1^k, \quad S_3^k = -g_3^k,$$

для зависимых (базисных) компонент вектора S^k найдем:

$$S_2^k = -A_2^{-1}(A_1 g_1^k + A_3 g_3^k).$$

Таким образом, при $\lambda_{k-1} = \lambda''_{k-1} < \lambda'_{k-1}$ направление S^k определено: мы проектируем антиградиент на гиперплоскости, ортогональные разности $(\ell+1)$ -го и ℓ -го направлений.

В обоих случаях определения возможного подходящего направления вычисляется скалярное произведение $gs = (g^k)^T S^k$.

Итак, кратко алгоритм выглядит следующим образом (подробности см. в блок-схемах и автокодовой программе).

1°. Пусть определены итерационные точки $x^0, x^1, \dots, x^k$

$$(x^h \in X, F(x^h) < F(x^{h-1})).$$

2°. Вычисляем g^* .

2а. Если $\lambda_{k-1} = \lambda'_{k-1}$, то полагаем $S_i^* = -g_i^*$, $i=1, \dots, m$. При этом $g_i^* = 0$, если направление $-g_i^*$ не является возможным.

2б. Если $\lambda_{k-1} = \lambda''_{k-1} < \lambda'_{k-1}$, то учитываются условия сопряженности (о преобразовании матрицы A см. пояснения к блок-схеме).

3°. Вычисляем $gs = (g^*)^T S^*$. Если gs мало, процесс прекращается.

4°. Определяем λ'_k и λ''_k .

5°. Находим $\lambda_k, x^{k+1} = x^k + \lambda_k S^*$ и вычисляем $F(x^{k+1})$ и δF . Если δF мало, процесс прекращается, в противном случае возвращаемся к 2°.

4. Блок-схема программы

На рис. 1 изображена общая блок-схема программы. В прямоугольниках, обведенных одной чертой, показаны простые (или составные) операторы; прямоугольники, обведенные двойной чертой, соответствуют целым блокам, которые на последующих рисунках (2-14) расшифровываются детально. В ромбиках записано логическое условие, истинность которого подлежит проверке. В квадратиках выделены метки некоторых операторов.

В операторах присваивания используется знак " $=$ ". Для операторов цикла принято обозначение: $i = i + \Delta$; a, b . Эта запись означает, что параметр цикла i изменяется от a до b с шагом Δ . Операторы цикла показаны не везде: там, где это можно, приведены операции над векторами (например, $g' = \bar{g}$). Подпрограммы обозначены так же, как блоки.

Как уже говорилось, программа предназначена для решения как динамических, так и статических задач оптимизации. Нестандартным блоком для динамического случая является подпрограмма счета правых частей (ПОД $SP2$) для системы (1), для статической задачи нестандартный блок - подпрограмма вычисления функции $F(x)$ ставится на место ПОД F , ПОД $AWJN$, ПОД $RUKU$, ПОД $SP2$ и ПОД RNU с помощью корректировочного оператора ЗАМЕНИТЬ.

Переменные y стандартизованы таким образом, что значение целевой функции присваивается переменной $y_{k+1|1}$, значение времени t - переменной $y_{k+1|2}$.

Мы уже отмечали, что программа позволяет решать задачи как с гладкими целевыми функциями, так и с негладкими. Признаком того, что задача имеет гладкую функцию цели, является задание $\epsilon < \epsilon_1$ (противоположное условие должно быть задано для задач с негладкими функциями цели).

Пояснения к общей блок-схеме. После ввода исходных данных проверяется, является ли задача "гладкой" или нет. Для "гладкой" решение проводится по способу сопряженных направлений, для "негладкой" - по методу конфигураций [12]. Перед работой по первому способу вычисляются величина масштаба (переменная MAS), необходимого для определения относительной погрешности.

Вся последующая процедура понятна из блок-схемы и описания алгоритма.

Пояснения к отдельным блокам

Блок конфигураций (рис. 2-4) отличается от соответствующей блок-схемы в [12] тем, что, во-первых, после осуществления результирующего шага проверяется, уменьшалась ли целевая функция, а потом уже производятся пробные шаги (этим достигается большая эффективность процесса), во-вторых, в процесс решения включены ограничения, отсутствующие в [12].

Блок вычисления градиента (рис. 5) определяет частные производные от целевой функции по параметрам так же, как и в [1], по формулам численного дифференцирования.

Блок сопряженных направлений (рис. 6) вычисляет взаимное подходящее направление с учетом условий сопряженности. Как уже отмечалось, должно выполняться условие $\rho \leq m-1$. В начале блока проверяется выполнение этого условия, и, если это возможно, ρ увеличивается. Если $\rho \geq 2$,

то строки матрицы A сдвигаются вниз: первая на место второй, ..., $(p-1)$ -я на место p -ой. Первая строка (если $p=1$, то она единственная) формируется как разность k -го и $(k-1)$ -го направлений.

Далее квадратная подматрица размерности $p \times p$ переносится в рабочий массив AR , с которым будет производиться процедура обращения, и запоминается $p1$ - номер первого столбца этой подматрицы. Если в процессе обращения матрицы AR окажется, что она вырожденная ($N4=0$ - см. ПОД ОБМАТ), то в массив AR выбирается из A следующая подматрица $p \times p$, т.е. $p1$ увеличивается на 1, если это возможно. Если все соседние подматрицы размерности $p \times p$ окажутся вырожденными, то p -число строк матрицы A - уменьшается на 1, если это возможно, и процедура обращения повторяется. При этом, если в матрице A не остается ни одной строки, то очередной итерационный шаг производится без учета условий сопряженности.

После обращения матрицы $AR=A2$ вычисляются базисные компоненты вектора S^* (внебазисные сохраняют значений компонент антиградиента), и подсчитывается величина qs .

Б л о к и λ' (рис.7) и λ'' (рис.8) вычисляют значения λ'_k , λ''_k , λ_k , а также находят следующую итерационную точку x^{k+1} и значение в ней целевой функции.

Обращение матрицы производится по способу Рунтисхаузера [13]. Этот способ представляет собой модификацию метода Гаусса-Жордана, удобную для реализации на ЭЦМ. Весь процесс обращения по Рунтисхаузеру подразделяется на три этапа: 1) выбор главного элемента и запоминание его координат (блок-схема на рис.9); 2) вычисление векторов B и C и пересчет матрицы $A2''$; $\nu = 1, \dots, p$ (рис.10); 3) перестановка столбцов (рис.11) и строк (блок-схема аналогична рис.11) для получения $A2^{-1}$. Более подробных пояснений к этой подпрограмме мы не приводим.

Б л о к и н т е г р и р о в а н и я системы по методу Рунге-Кутты с автоматическим выбором шага представлен четырьмя подпрограммами:

1) ПОД F (рис. 12), где предусмотрена возможность получения начальных условий по некоторым координатам;

2) ПОД $RUKU$ (рис. 13), осуществляющей собственно интег-

рирование;

3) ПОД $AWJN$ (рис. 14), в которой происходит выбор шага интегрирования по величине заданной абсолютной погрешности;

4) ПОД $SP4$, где производится счет правых частей системы.

Полный текст автокодовой программы УПОЭЦ (без тела ПОД $SP2$, добавленного к программе с помощью корректировочного оператора ВСТАВИТЬ) приведен в приложении.

5. Примеры

В качестве тестового примера для отладки программы УПОЭЦ и проверки ее эффективности была выбрана функция

$$F(x) = x_1^2 + 5x_2^2 + 100x_3^2, \quad (10)$$

линии уровня которой представляют собой сильно вытянутые эллипсы.

Применение метода наискорейшего спуска для отыскания минимума этой функции - точки $\hat{x}^T = \|0,0,0\|$ - с заданной точностью дает медленную сходимость (см. табл. I). При этом следует заметить, что как по способу вычисления градиента, так и по способу определения минимума по направлению в применяемом методе, во-первых, не производится "линейных" вычислений $F(x)$, во-вторых, и градиент, и минимум по направлению для квадратичной функции определяются точно (в пределах точности машины).

Для сравнения в той же табл. I приведены результаты минимизации функции (10) с помощью программы УПОЭЦ, показывающие преимущество метода сопряженных направлений (по сравнению с методом наискорейшего спуска).

Вторым примером на применение программы УПОЭЦ может служить задача оптимизации параметров критического автомата [15,2] по критерию быстрогодействия, когда целевой функцией является функционал типа Майера:

$$T = t_N - t_0. \quad (11)$$

Таблица 1

Методы	Число шагов	$\hat{x}$	$\hat{J}$
Наискорейшего спуска	711	$0,944 \cdot 10^{-3}$ $0,838 \cdot 10^{-8}$ $-0,285 \cdot 10^{-5}$	$0,892 \cdot 10^{-6}$
Сопряженных направлений (программа УПОЭЦ)	41	$-0,135 \cdot 10^{-7}$ $-0,167 \cdot 10^{-5}$ $-0,254 \cdot 10^{-6}$	$0,203 \cdot 10^{-10}$
минимум		0 0 0	0

Уравнения криотронного автомата даны в [15]. Здесь мы кратко опишем результаты решения этой оптимальной задачи.

Таблица 2

№	x_1^*	x_2^*	x_3^*	y_1^*	y_2^*	T^*	$\hat{x}_1$	$\hat{x}_2$	$\hat{x}_3$	$\hat{T}$
1	0,3	0,7	1	1	0,25	10	0,3	0,7	0,4	4
2	0,3	0,7	1	0	0,55	10	0,3	0,7	0,4	4
3	0,4	0,75	0,85	0	0,75	10	0,4	0,764	0,4	4
4	0,4	0,75	0,85	1	0,25	8,5	0,4	0,75	0,4	4

В табл. 2 приведены исходные данные и результаты счета для четырех вариантов. На рис. 15-18 изображены соответствующие кривые переходных процессов.

Следует отметить, что критерий оптимизации по качеству переходного процесса, принятый в [2], оказался более "чувствительным" к параметрам x_1 и x_2 , чем применяемый здесь (II).

ЛИТЕРАТУРА

1. В.К. КОРОЛЕВ. Применение методов квадратичного программирования для оптимизации параметров динамических систем. Данный сборник.
2. В.К. КОРОЛЕВ. Оптимизация криотронного автомата. Данный сборник.
3. В.К. КОРОЛЕВ. Расчет оптимальных параметров импульсного трансформатора. -Вычислительные системы, Новосибирск, "Наука" СО, 1967, вып. 27.
4. В.К. КОРОЛЕВ. Оптимизация параметров усилителя-формирователя. -Вычислительные системы, Новосибирск, "Наука" СО, 1967, вып. 27.
5. Г.М. ФИХТЕНГОЛЬЦ. Курс дифференциального и интегрального исчисления. т.1, Физматгиз, М., 1962.
6. Л.С. ПОНТРИГИН. Обыкновенные дифференциальные уравнения, "Наука" М., 1965.
7. Р.Е. FLEISCHER. Optimization Techniques in System Design, System Analysis by Digital Computer, New Jersey, May, 1966.
8. Л.Я. ЦИЛБ. Вариационное исчисление и интегральные уравнения, "Наука", М., 1966.
9. Г. ЗОЛТЕНДЕН. Методы возможных направлений. ИЛ, М., 1963.
10. Н.С. БЕРЕСИН, Н.П. ИВАНОВ. Методы вычислений. т.1, Физматгиз, М., 1962.
11. R. FLEISCHER and M.J.D. POWELL. A rapidly convergent descent method for minimization. Computer J., vol.6, 1963.
12. R. HOOKE and T.A. JEEVES. "Direct search" solution of numerical and statistical problems. Journ. of Assoc. Comp. Mach., vol.8, N 2, 1961.
13. Н. ВУТЕНБАУСЕР. Zur Matrizeninversion nach Gauss-Jordan, ZAMP, X, 1959.
14. Д.Д. УАЙЛД. Методы поиска экстремума, "Наука", М., 1967.
15. Б.М. СОМОВ. Вопросы динамики асинхронных автоматов. -Вычислительные системы, Новосибирск, "Наука" СО, 1968, вып. 33.

Поступила в редакцию
14.1.1969 г.

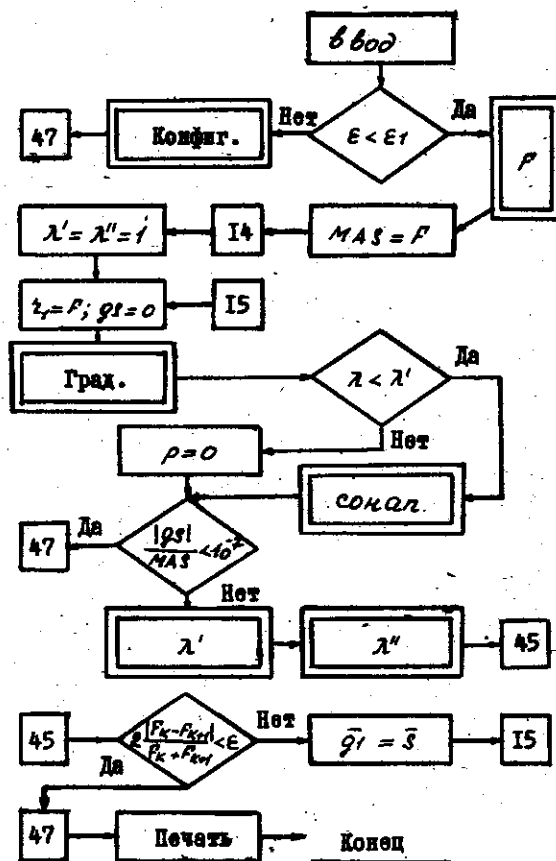


Рис. 1

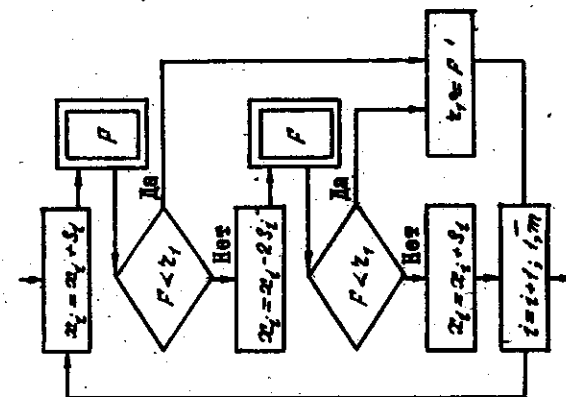


Рис. 3

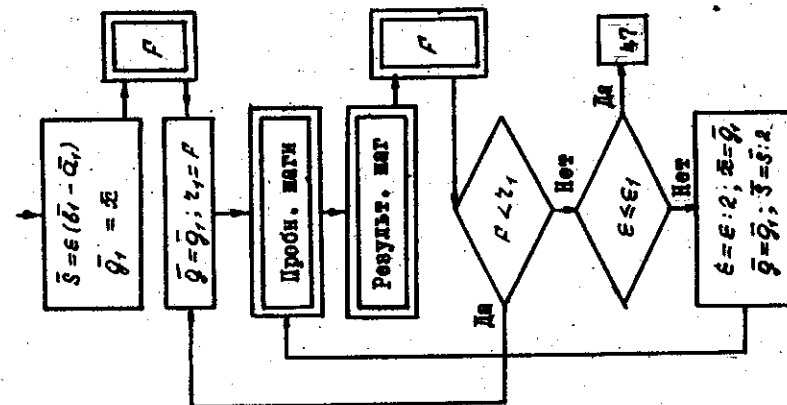


Рис. 2

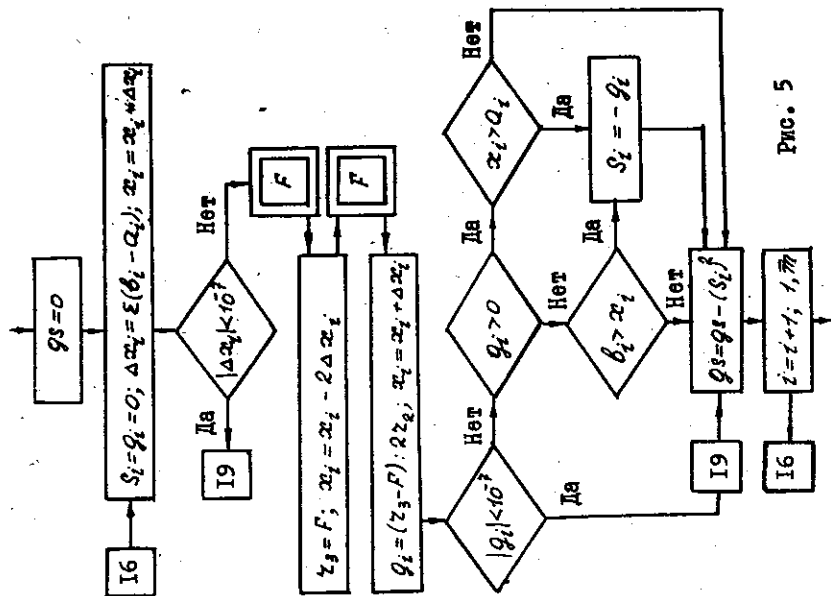


Рис. 5

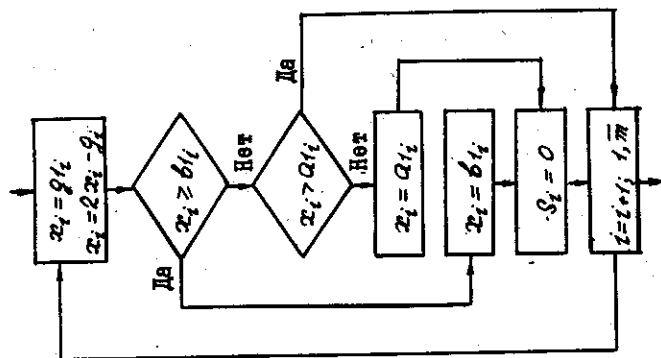


Рис. 4

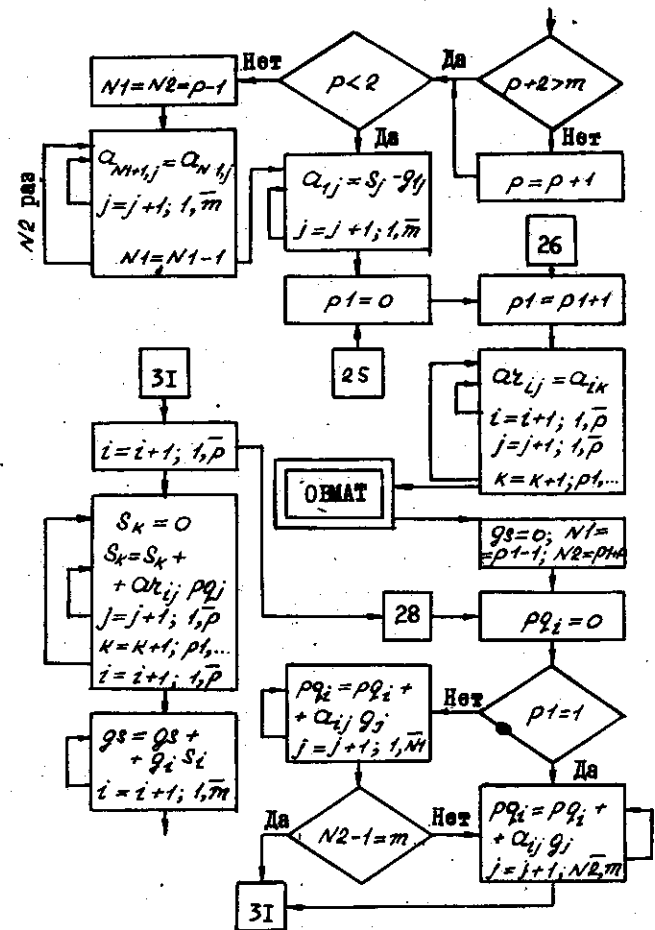
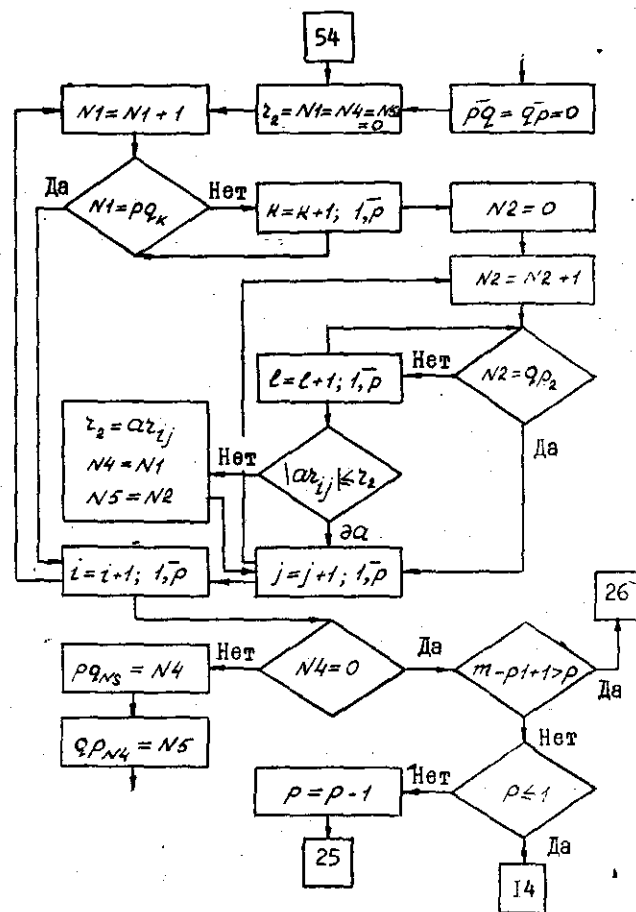
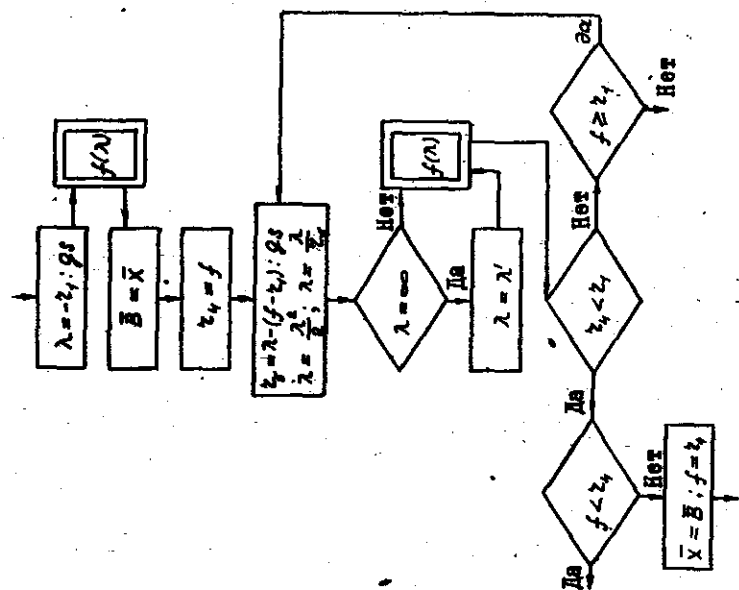


Рис. 6



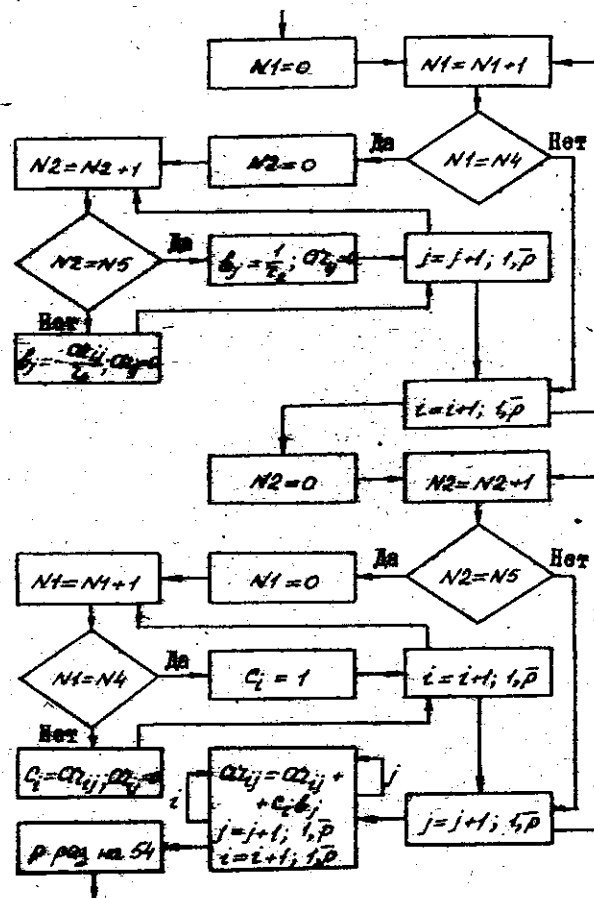


Рис. 10

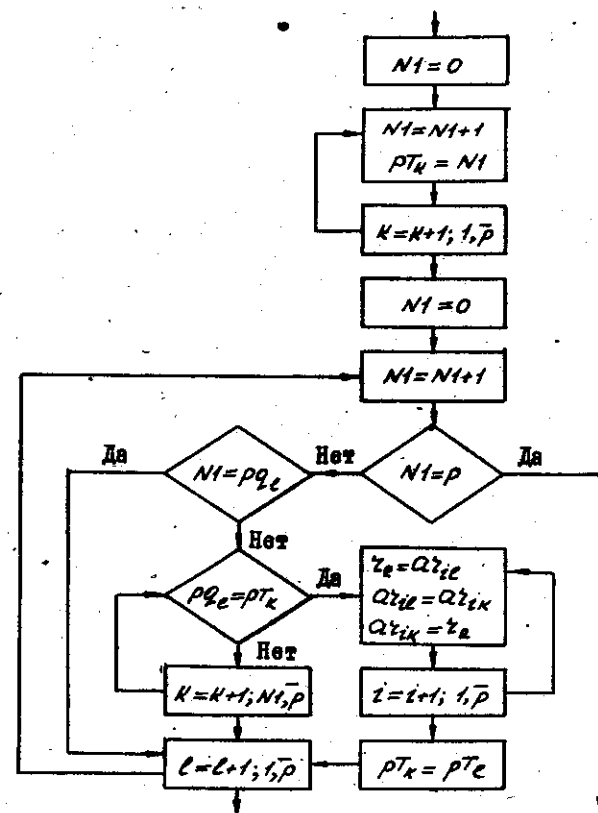


Рис. 11

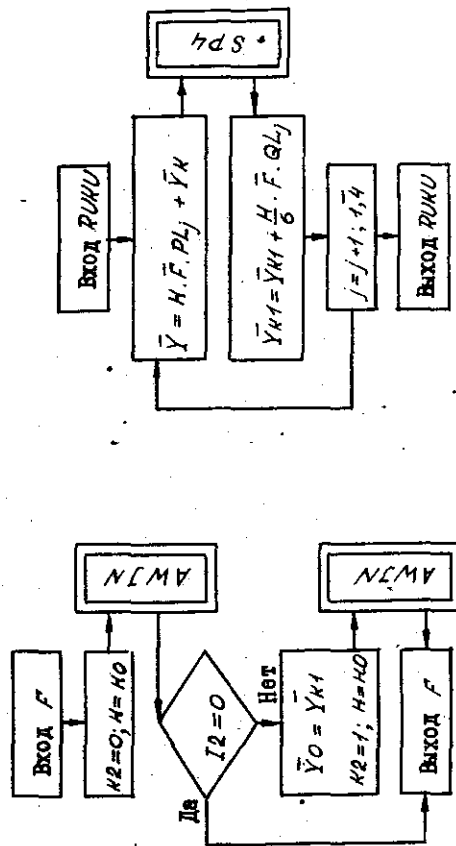


Рис. 12

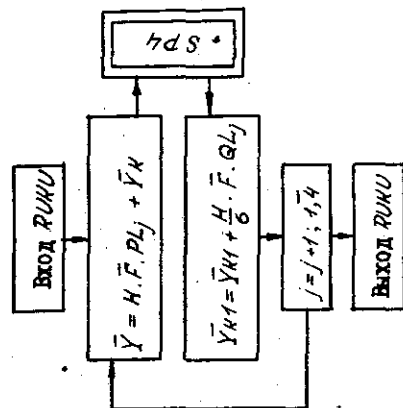


Рис. 13

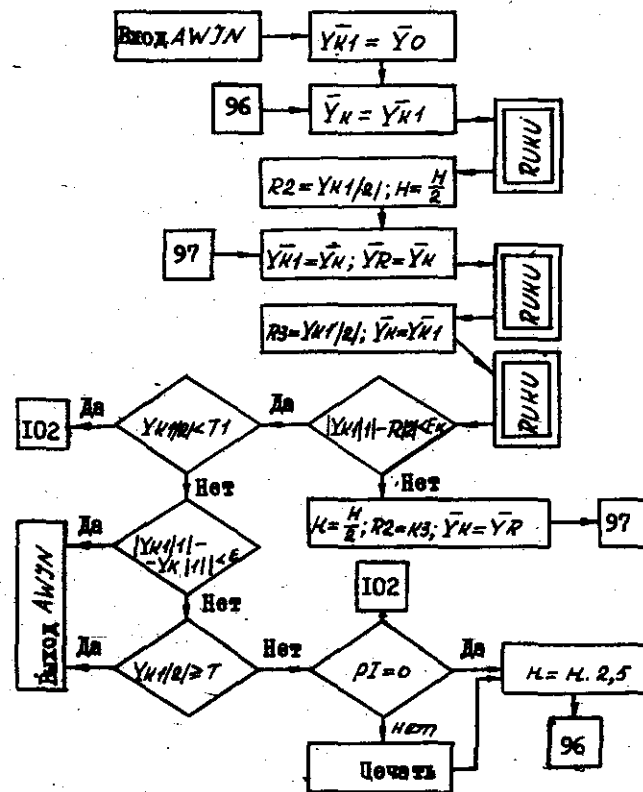


Рис. 14.

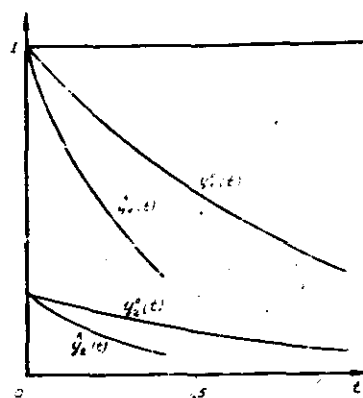


Рис. 15

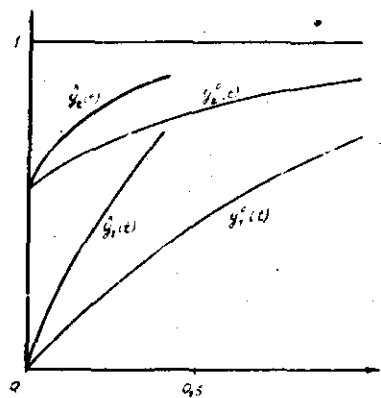


Рис. 16

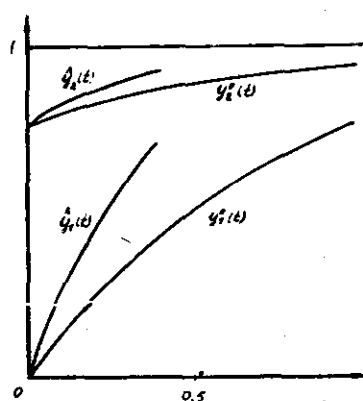


Рис. 17

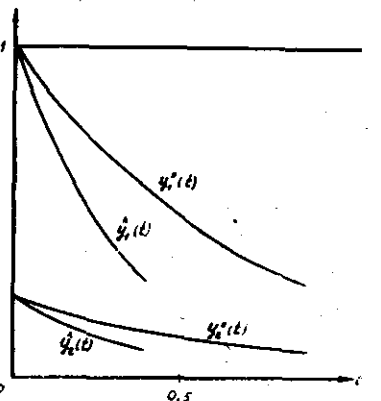


Рис. 18

```

01  УПОСЫ
02  MAC A(870 29.30), AR(870 29.30), G(30), S(30),
03  PQ(30), QP(30), PT(30), X1(30), B(30), C(30)X
04  НАЗ PI=0.0, 5.0, 5.1 QI=1.2.2.1 YR(90)=AR/1.1/
05  Y(90)=AR/4.1/ YK(90)=AR/7.1/ YK1(9)=AR/10.1/
06  P(90)=AR/13.1/ G1(30)=AR/16.1/X
07  1. ВВОД E, E1, E2, E3, EK, HO, T, T1, A1(30), X(30), B1(30), YO(9):N,
    M, I1, I2X
08  ВЫЧ :PIK=OX
09  2. ВЫЧ :PI=1X
10  ВЫП 91X
11  НАП NA BPM YK1/1/, X(30)X
12  ЕСЛ :PIK =0 TO 17X
13  КОН X
14  17. ВЫЧ :PI=OX
15  ЕСЛ E (E1 TO 13X
16  3. ВЫЧ S/I=E. (B1/I/-A1/I/) G1/I=X/I/X

```

```

01  ПОВ 3 I=1 (1) MX
02  4. ВЫЧ G/I=G1/I/X
03  ПОВ 4 I=1 (1) MX
04  ВЫЧ R1=YK1/1/X
05  5. ВЫЧ R2=S/I/ X/I/=X/I/+R2X
06  ВЫП 91X
07  ЕСЛ YK1/1/ (R1 TO 6X
08  ВЫЧ X/I=X/I/-2.R2X
09  ВЫП 91X
10  ЕСЛ YK1/1/ (R1 TO 6X
11  ВЫЧ X/I=X/I/+R2X
12  ПЕР 7X
13  6. ВЫЧ R1=YK1/1/X
14  7. ПОВ 5 I=1 (1) MX
15  8. ВЫЧ G1/I=X/I/ X/I=X/I/.2-G/I/X
16  ЕСЛ X/I/ )=B1/I/ TO 9X

```

```

01  ЕСЛ X/I/ )A1/I/ TO 11X
02  ВЫЧ X/I/=A1/I/X
03  ПЕР 10X
04  9. ВЫЧ X/I/=B1/I/X
05  10. ВЫЧ S/I=OX
06  11. ПОВ 8 I=1 (1) MX

```

07 BNP 91X/
08 ECI YK1/1/ (R1 TO 4X
09 ECI E (=R1 TO 48X
10 BNP E=E:2X
11 12. BNP X/I/=G1/I/ G/I=G1/I/ S/I=S/I/:2X
12 NOB 12 I=1 (1) MX
13 NEP 5X
14 13. BNP MAS=YK1/1/X
15 14. BNP LAM=1 LAM1=1X
16 15. BNP R1=YK1/1/ GS=0X

INCT 04

01 HAN NA BFM R1X
02 16. BNP X1/I/=X/I/ S/I=0 G/I=0X
03 BNP E4=X/I/X
04 HAN NA BFM R4X
05 BNP R2=X. (B1/I/-A1/I/) X/I=X/I/+R2X
06 ECI MOD(R2) (D-7 TO 19X
07 BNP 91X
08 BNP R3=YK1/1/ X/I=X/I/-2.R2X
09 BNP 91X
10 BNP R5=(R3-YK1/1/):2.R2 X/I=X/I/+R2 G/I=R5X
11 ECI MOD(R5) (L-7 TO 19X
12 ECI R5)0 TO 18X
13 ECI B1/I/)X/I/ TO 35X
14 NEP 19X
15 18. ECI X/I/)A1/I/ TO 35X
16 NEP 19X

INCT 05

01 35. BNP S/I/=-R5X
02 19. BNP GS=GS-S/I/'2X
03 NOB 16 I=1 (1) MX
04 BNP GS1=GSX
05 ECI LAM (LAM1 TO 21X
06 BNP :P=0X
07 20. ECI MOD(GS:MAS) (D-7 TO 47 INAVE 36X
08 ECI :P+2)M TO 22X
09 BNP :P=P+1X
10 ECI :P (2 TO 24X
11 BNP :N1=P-1 N2=N1X
12 23. BNP A/I+1, J/=A/I, J/X
13 NOB 23 J=1 (1) MX
14 NOB 23 I=N1 (1).1X
15 BNP :N1=N1-1X
16

INCT 06

01 NOB 23 N2X
02 24. BNP A/1, J/=S/J/-G1/J/X
03 NOB 24 J=1 (1) MX
04 25. BNP :P1=0X
05 26. BNP :P1=P1+1X
06 27. BNP AR/I, J/=A/I, K/X
07 NOB 27 I=1 (1) PX
08 NOB 27 J=1 (1).K=P1 (1).PX
09 BNP 52X
10 BNP GS=0 GS2=0X
11 BNP :N1=P1-1 N2=P1+PX
12 28. BNP PQ/I/=0X
13 ECI :P1 =1 TO 30X
14 29. BNP PQ/I/=PQ/I/+A/I, J/.G/J/X
15 NOB 29 J=1 (1) N1X
16 ECI :N2-1 =M TO 31X

INCT 07

01 30. BNP PQ/I/=PQ/I/+A/I, J/.G/J/X
02 NOB 30 J=N2 (1) MX
03 31. NOB 28 I=1 (1) PX
04 32. BNP S/K/=0X
05 33. BNP S/K/=S/K/+AR/I, J/.PQ/J/X
06 NOB 33 J=1 (1) PX
07 NOB 32 K=P1 (1).I=1 (1) PX
08 34. BNP GS2=GS2+S/I/'2X
09 NOB 34 I=1 (1) MX
10 BNP RM=(GS1:GS2)' (1:2)X
11 41. BNP S/I/=S/I/, RM GS=GS+G/I/.S/I/X
12 NOB 41 I=1 (1) MX
13 ECI GS (0 TO 20X
14 42. BNP S/I/=-S/I/X
15 NOB 42 I=1 (1) MX
16

INCT 08

01 BNP GS=-GSX
02 NEP 20X
03 36. BNP LAM1=D10X
04 37. BNP R2=S/I/ R3=A1/1/-X/I/ R4=B1/I/-X/I/X
05 ECI R2)0 TO 38X
06 BNP LAM=R3 LAM=LAM:R2X
07 KOD -3300 0047 0050.-3077 0000 0000X
08 38. BNP LAM=R4 LAM=LAM:R2X
09 KOD -3300 0047 0050.-3077 0000 0000X
10 39. ECI LAM)=LAM1 TO 40X
11 BNP LAM1=LAMX

12 40. NOB 37 L=1 (1) MX
13 BMQ LAM=R1:GSX
14 BMH 49X
15 43. BMQ R/I=X/I/X
16 NOB 43 L=1 (1) MX

INCT 09

01 BMQ R4=YK1/1/X
02 44. BMQ R5=LAM-(YK1/1/-R1):GS LAM=LAM'2=3
03 LAM=LAM:R5X
04 KOX -3300 0155 0156.-3077 0000 0000X
05 109. BMQ LAM=LAM1X
06 110. BMH 49X
07 ECH R4 (R1 TO 111X
08 ECH YK1/1/)=R1 TO 44 INA4E 45X
09 111. ECH YK1/1/ (R4 TO 45X
10 112. BMQ X/I=X/I/X
11 NOB 112 L=1 (1) MX
12 BMQ YK1/1/=BAX
13 45. ECH 2. (R1-YK1/1/):(R1+YK1/1/) (R2 TO 47X
14 46. BMQ C/I/I=X/I/X
15 NOB 46 L=1 (1) MX
16 BEP 15X

INCT 10

01 47. ECH TP 0 TO 14X
02 48. BMQ :PIK=1X
03 BEP 2X
04 49. NOB XX
05 ECH LAM)=0 TO 108X
06 BMQ LAM=OX
07 108. ECH LAM (LAM1 TO 50X
08 BMQ LAM=LAM1X
09 50. BMQ X/I=X/I/I/+LAM.S/I/X
10 NOB 50 L=1 (1) MX
11 BMH 91X
12 BMX XX
13 52. NOB OBHMAX
14 53. BMQ :PQ/I/=0 QP/I/=OX
15 NOB 53 L=1 (1) PX
16 54. BMQ R2=OX

INCT 11

01 BMQ :N1=0 N4=0 N5=OX
02 55. BMQ :N1=N1+1X
03 56. ECH :N1=PQ/K/ TO 60X

04 NOB 56 K=1 (1) PX
05 BMQ :N2=OX
06 57. BMQ :N2=N2+1X
07 58. ECH :N2=QP/I/ TO 59X
08 NOB 58 L=1 (1) PX
09 ECH MOD(AR/I,J/) (=R2 TO 59X
10 BMQ R2=AR/I,J/X
11 BMQ :N4=N1 N5=N2X
12 59. NOB 57 J=1 (1) PX
13 60. NOB 55 L=1 (1) PX
14 ECH :N4=0 TO 61 INA4E 62X
15 61. ECH :M=P+1)P TO 26X
16 ECH :P (=1 TO 14X

INCT 12

01 BMQ :P=P-1X
02 BEP 25X
03 62. BMQ :PQ/K/=N4X
04 NOB 62 K=N5 (1).1X
05 63. BMQ :QP/I/=N5X
06 NOB 63 L=N4 (1).1X
07 BMQ :N1=OX
08 64. BMQ :N1=N1+1X
09 ECH :N1=N4 TO 65 INA4E 69X
10 65. BMQ :N2=OX
11 66. BMQ :N2=N2+1X
12 ECH :N2=N5 TO 67X
13 BMQ R/J=-AR/I,J/:R2 AR/I,J/=OX
14 BEP 68X
15 67. BMQ R/J=:1:R2 AR/I,J/=OX
16 68. NOB 66 J=1 (1) PX

INCT 13

01 69. NOB 64 L=1 (1) PX
02 BMQ :N2=OX
03 70. BMQ :N2=N2+1X
04 ECH :N2=N5 TO 71 INA4E 75X
05 71. BMQ :N1=OX
06 72. BMQ :N1=N1+1X
07 ECH :N1=N4 TO 73X
08 BMQ C/I/=AR/I,J/ AR/I,J/=OX
09 BEP 74X
10 73. BMQ C/I/=1X
11 74. NOB 72 L=1 (1) PX
12 75. NOB 70 J=1 (1) PX
13 76. BMQ AR/I,J/=AR/I,J/+C/I/.R/J/X
14 NOB 76 J=1 (1) PX
15 NOB 76 L=1 (1) PX
16 NOB 54 PX

ЛИСТ 14

```

01      БМЧ :N1=0X
02 77. БМЧ :N1=N1+1 PT/K/=N1X
03      ПОВ 77 K=1 (1) PX
04      БМЧ :N1=0X
05 78. БМЧ :N1=N1+1X
06      ЕСЛ :N1 =P TO 83X
07      ЕСЛ :N1 =PQ/L/ TO 82X
08 79. ЕСЛ :PQ/L/ =PT/K/ TO 80 INA4E 81X
09 80. БМЧ R2=AR/I,L/ AR/I,L/=AR/I,K/ AR/I,K/=R2X
10      ПОВ 80 I=1 (1) PX
11      БМЧ :PT/K/=PT/L/X
12      ПЕР 82X
13 81. ПОВ 79 K=N1 (1) PX
14 82. ПОВ 78 I=1 (1) PX
15 83. БМЧ :N1=0X
16 84. БМЧ :N1=N1+1 PT/K/=N1X

```

ЛИСТ 15

```

01      ПОВ 84 K=1 (1) PX
02      БМЧ :N1=0X
03 85. БМЧ :N1=N1+1X
04      ЕСЛ :N1 =P TO 90X
05      ЕСЛ :N1 =QP/L/ TO 89X
06 86. ЕСЛ :QP/L/ =PT/K/ TO 87 INA4E 88X
07 87. БМЧ R2=AR/L,J/ AR/L,J/=AR/K,J/ AR/K,J/=R2X
08      ПОВ 87 J=1 (1) PX
09      БМЧ :PT/K/=PT/L/X
10      ПЕР 89X
11 88. ПОВ 86 K=N1 (1) PX
12 89. ПОВ 85 I=1 (1) PX
13 90. БМЧ OBMATX
14 91. ПОД FX
15      БМЧ K2=0 H=HOX
16      БМЧ 94X

```

ЛИСТ 16

```

01      ЕСЛ :I2 =0 TO 93X
02 92. БМЧ Y0/I/=YK1/I/X
03      ПОВ 92 I=I1 (1) I2X
04      БМЧ K2=1 H=HOX
05      БМЧ 94X
06 93. БМЧ FX
07 94. ПОД AWINX
08 95. БМЧ YK1/I/=Y0/I/X
09      ПОВ 95 I=1 (1) NI
10 96. БМЧ YK/I/=YK1/I/X

```

```

11      ПОВ 96 I=1 (1) NX
12      БМЧ 104X
13      БМЧ R2=YK1/1/ H=H:2X
14 97. БМЧ YK1/I/=YK/I/ YR/I/=YK/I/X
15      ПОВ 97 I=1 (1) NX
16      БМЧ 104X

```

ЛИСТ 17

```

01      БМЧ R2=YK1/1/X
02 98. БМЧ YK/I/=YK1/I/X
03      ПОВ 98 I=1 (1) NX
04      БМЧ 104X
05      ЕСЛ MOD(YK1/1/-R2) (ЕК TO 101X
06      БМЧ H=H:2 R2=R3X
07 100. БМЧ YK/I/=YR/I/X
08      ПОВ 100 I=1 (1) NX
09      ПЕР 97X
10 101. ЕСЛ YK1/2/ (T1 TO 102X
11      ЕСЛ MOD(YK1/1/-YK/1/) (ЕК TO 103X
12      ЕСЛ YK1/2/ )=T TO 103X
13 102. ЕСЛ :PI =0 TO 99X
14      НАП NA BPM YK1/2/,Z9X
15 99. БМЧ H=H:2,5X
16      ПЕР 96X

```

ЛИСТ 18

```

01 103. БМЧ AWINX
02 104. ПОД RUKUX
03 105. БМЧ Y/I/=H.F/I/.FL/J/+YK/I/X
04      ПОВ 105 I=1 (1) NX
05      БМЧ 107X
06 106. БМЧ YK1/I/=YK1/I/+H:6.F/I/.QL/J/X
07      ПОВ 106 I=1 (1) NX
08      ПОВ 105 J=1 (1) 4X
09      БМЧ RUKUX
10 107. ПОД SP4X
11      БМЧ SP4X
12      НАЧ 1X

```