

УДК 518.5:681.142.001.2

ПРИМЕНЕНИЕ МЕТОДОВ КВАДРАТИЧНОГО ПРОГРАММИРОВАНИЯ ДЛЯ ОПТИМИЗАЦИИ ПАРАМЕТРОВ ДИНАМИЧЕСКИХ СИСТЕМ

В.К. Королев

Излагается методика расчета оптимальных параметров динамических систем, основанная на применении методов квадратичного программирования. Дается описание универсальной программы оптимизации. Приводится пример расчета для одного из элементов вычислительной техники.

I. Постановка задачи

Рассматривается некоторая динамическая система, описываемая следующими уравнениями:

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

с начальными условиями: $y(t_0) = y^0$.
Здесь $y^T = \|y_1, \dots, y_n\|$ и $x^T = \|x_1, \dots, x_p\|$ - векторы состояний системы, $z^T = \|z_1, \dots, z_m\|$ - вектор параметров системы.

*) Например, токи, напряжения и т.п.

**) Например, сопротивления, емкости и т.п.

t - время (значок "T" означает транспонирование).

Система (I) рассматривается на конечном интервале времени T :

$$T = \{t | t_0 \leq t \leq t_n\}.$$

На параметры системы наложены двусторонние ограничения так, что область их изменения есть m -мерный параллелепипед X :

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

Система конечных уравнений в (I) предполагается разрешимой относительно вектора x для каждого момента времени $t \in T$ и для всех допустимых значений параметров $x \in X$, хотя это не всегда возможно сделать явно. Векторные функции φ и f предполагаются дважды непрерывно дифференцируемыми по x , y и z .

Рассматривается функционал:

$$S[x] = \int_{t_0}^{t_n} [x(x, y, z, t) - x_*(t)]^2 dt, \quad (3)$$

представляющий квадрат отклонения "выходной" функции системы x , от заданной функции времени $x_*(t)$.

На решении системы (I) функционал (3) превращается в функцию параметров $S(x)$. Ставится задача отыскания её минимума в области (2):

$$\min \{S(x) | x \in X\}. \quad (4)$$

Некоторые замечания. Сформулированное выше есть задача нелинейного математического программирования с весьма простой (выпуклой) допустимой областью и сложным заданием функции цели.

Без специальных требований на правые части системы (I) вопрос о выпуклости целевой функции $S(x)$, значения которой могут быть получены, вообще говоря, только численно, остается открытым. Поэтому задача ставится на отыскание лишь локального минимума.

В [1] применен один из "инженерных" способов оптимизации, взятый из [3], в работе [2] приводится реализация более тонкого метода возможных направлений Зойтендейка [4]. Отправным

моментом при переходе от общего метода Зойтендейка к излагаемому ниже было стремление линейризовать трансцендентное (в общем случае) уравнение для определения величины градиентного шага. Далее, способ вычисления производных по параметрам от целевой функции, применявшийся в [1] и [2], был основан на интегрировании так называемой системы уравнений в вариациях, что не позволяло эффективно стандартизовать программу.

При численном способе дифференцирования оказывается возможным оставить в качестве нестандартной части программы лишь два блока: счета правых частей системы (I) и расчета начального значения функционала (3).

Кроме того, в данной работе остается вопрос об определении допусков на оптимальные значения параметров.

2. Сведение задачи (4) к задаче квадратичного программирования

Идея предлагаемого метода состоит в линейризации выходной функции по вариациям параметров в окрестности допустимой точки $x \in X$, минимизации в этой окрестности получающейся квадратичной формы для $S(x)$ и повторении этой процедуры в найденной точке минимума.

$$\text{Пусть} \quad \begin{cases} y^o = y(x^o, t), \\ z^o = z(x^o, t). \end{cases}$$

- решение системы (I) для допустимой точки x^o . Разложим выходную функцию x_i в ряд Тейлора по вариациям параметров:

$$x_i = x_i^o + \sum_j \frac{\partial x_i^o}{\partial x_j} \delta x_j + \frac{1}{2} \sum_j \sum_k \frac{\partial^2 x_i^o}{\partial x_j \partial x_k} \delta x_j \delta x_k + o(\delta x^2), \quad (5)$$

где

$$x_i^o = x_i(x^o, y^o, z^o, t),$$

$$\frac{\partial x_i^o}{\partial x_j} = \frac{\partial x_i}{\partial x_j} \bigg|_{(x^o, y^o, z^o, t)}, \quad (6)$$

$$\frac{\partial^2 x_i^o}{\partial x_j \partial x_k} = \frac{\partial^2 x_i}{\partial x_j \partial x_k} \bigg|_{(x^o, y^o, z^o, t)}.$$

Требование дважды непрерывной дифференцируемости правых частей (1) обеспечивает непрерывность вторых производных $\frac{\partial^2 z_i}{\partial x_i \partial x_j}$ [5]. Поэтому остаточный член в разложении (5) будет величиной более высокого порядка малости, чем δx^2 , и в дальнейшем опускается.

Подставляя линейную часть разложения в выражение функционала, мы получаем локальную аппроксимацию целевой функции квадратичной формой:

$$S(\delta x) = \int_{t_0}^{t_N} [z_i^* - z_* + \sum_i \frac{\partial z_i^*}{\partial x_i} \delta x_i]^2 dt = S^0 + P^T \delta x + \delta x^T C \delta x, \quad (7)$$

где

$$\begin{aligned} S^0 &= \int_{t_0}^{t_N} (z_i^* - z_*)^2 dt, \\ P_i &= 2 \int_{t_0}^{t_N} (z_i^* - z_*) \frac{\partial z_i^*}{\partial x_i} dt, \\ C_{ij} &= \int_{t_0}^{t_N} \frac{\partial z_i^*}{\partial x_i} \cdot \frac{\partial z_j^*}{\partial x_j} dt, \end{aligned} \quad (8)$$

(0 вычислении производных (6) см. п⁰4).

Для определения окрестности, в которой справедлива линеаризация функции z_i , потребуем, чтобы квадратичная часть разложения была величиной порядка ε , где $\varepsilon > 0$ — заданное малое число, по сравнению с линейными членами:

$$\| \frac{1}{2} \sum_i \sum_j \frac{\partial^2 z_i^*}{\partial x_i \partial x_j} \delta x_i \delta x_j \| \leq \varepsilon \| \sum_i \frac{\partial z_i^*}{\partial x_i} \delta x_i \|.$$

Отсюда определим $\Delta x_i \equiv |\delta x_i|$ следующим образом:

$$\Delta x_i = 2\varepsilon \frac{\alpha_i}{\alpha_{ii}}, \quad (9)$$

где

$$\alpha_i^2 \equiv \left\| \frac{\partial z_i^*}{\partial x_i} \right\|^2 = \int_{t_0}^{t_N} \left(\frac{\partial z_i^*}{\partial x_i} \right)^2 dt,$$

$$\alpha_{ii}^2 \equiv \left\| \frac{\partial^2 z_i^*}{\partial x_i^2} \right\|^2 = \int_{t_0}^{t_N} \left(\frac{\partial^2 z_i^*}{\partial x_i^2} \right)^2 dt.$$

Задачу минимизации квадратичной формы (7) будем решать в локальной допустимой области Δx , представляющей пересечение окрестности (9) с исходной допустимой областью X :

$$\Delta x = \{ \delta x_i \mid x_{li} \leq \delta x_i \leq x_{pi}, i=1, \dots, m \}, \quad (10)$$

где

$$x_{li} = \max(\alpha_i - x_i, -\Delta x_i),$$

$$x_{pi} = \min(b_i - x_i, \Delta x_i).$$

3. Решение задачи квадратичного программирования

Итак, мы имеем следующую задачу:

$$\min \{ P^T \delta x + \delta x^T C \delta x \mid \delta x \in \Delta x \},$$

для решения которой можно применить любой из методов квадратичного программирования [8].

Использование метода возможных направлений Зойтендейка дает накопление ошибки счета в матричных операциях, что приводит к отрицательным значениям квадратичной формы $\delta x^T C \delta x$. В результате при вычислении градиентного шага получается отрицательное значение, и процесс теряет смысл. Поэтому применяется другой метод квадратичного программирования Хилдрета и д'Эзопо [8].

В силу того, что все $C_{ii} \geq 0$ (см. (8)), т.е. в сечении квадратичной формы гиперплоскостями, параллельными координатным, все параболы выпуклы вниз, полагаем на $(k+1)$ -й итерации:

$$\delta x_i^{k+1} = \begin{cases} x_{li} & , \text{ если } \tilde{\delta x}_i^{k+1} \leq x_{li}; \\ \tilde{\delta x}_i^{k+1} & , \text{ если } x_{li} \leq \tilde{\delta x}_i^{k+1} \leq x_{pi}; \\ x_{pi} & , \text{ если } x_{pi} \leq \tilde{\delta x}_i^{k+1}, \end{cases}$$

где

$$\delta x_i^{k+1} = \frac{1}{c_{ii}} \left\{ -P_i - \sum_{j=0}^{i-1} C_{ij} \delta x_j^{k+1} - \sum_{j=i+1}^m C_{ij} \delta x_j^k \right\}, i=1, \dots, m.$$

Итерации повторяются, пока все разности $|\delta x_i^{k+1} - \delta x_i^k|, i=1, \dots, m$, не сделаются достаточно малыми.

После того, как в области (10) найдена минимальная точка, она занимает место точки x^0 , и весь процесс повторяется. В результате мы приходим либо к внутреннему минимуму функции $S(x)$, что определяется по экстремуму аппроксимирующей квадратичной формы, либо выходим в такую граничную точку исходной допустимой области, откуда невозможно продолжить описанный выше процесс минимизации.

Допуски Δx_i на оптимальные значения параметров определяются следующим образом. Пренебрегая квадратичным членом функции $S(\delta x)$ в локальной допустимой области, потребуем, чтобы линейный член был порядка ε по сравнению с S^0 :

$$|P^T \Delta \bar{x}| \leq \varepsilon S^0.$$

Кроме того, величина допуска не должна превышать размер локальной допустимой области. Отсюда, используя принцип равных влияний [7], находим:

$$\Delta x_i = \min \left(\frac{\varepsilon S^0}{m |P_i|}, \Delta x_i \right), i=1, \dots, m.$$

4. Универсальная программа

Программа составлена на языке АКИ [9], [10] для ЭЦМ "Минск-2" и позволяет решать задачи типа (1)-(4) следующих размеров: $m \leq 20, n \leq 20$.

Она состоит из четырех функциональных блоков и трех подпрограмм, одна из которых содержит свою подпрограмму (рис.1)*:

- 1 - блок выбора шага интегрирования,
- 2 - блок счета коэффициентов квадратичной формы,
- 3 - блок решения квадратичной задачи,
- 4 - блок управления итерациями;

* Все рисунки даны в приложении.

- 5 - подпрограмма вычисления интеграла,
- 6 - подпрограмма интегрирования системы (1),
- 7 - подпрограмма счета правых частей системы (1),
- 8 - подпрограмма расчета начального значения функционала.

На рис.2-5 приведены подробные блок-схемы основных блоков программы. Поэтому мы ограничимся здесь лишь некоторыми пояснениями.

Б л о к 1 (рис. 2). Интегрирование системы (1) проводится методом Рунге-Кутты четвертого порядка с постоянным шагом, выбор величины которого существен: малая его величина ведет к расходованию излишнего машинного времени, большая - к потере точности счета.

Оптимальным шагом интегрирования h мы считаем такой, при котором выполнены следующие два соотношения:

$$\frac{|S_1 - S_2|}{\frac{1}{2}(S_1 + S_2)} > \varepsilon, \quad \frac{|S_1 - S_2|}{\frac{1}{2}(S_1 + S_2)} \leq \varepsilon, \quad (II)$$

где S_1, S_2 и S_2 - значения функционала (3), вычисленные при величинах шага соответственно $h, 2h, \frac{h}{2}$.

Блок 1 проводит интегрирование системы (1), последовательно уменьшая или увеличивая h и запоминая то его значение, при котором выполняется соотношение (II).

Так как для массива решения $\{z_1(t_0), z_1(t_1), \dots, z_1(t_N)\}$ в программе отведено определенное количество ячеек, то при дроблении шага число компонент данного массива может превысить размер отведенного для него места. В таком случае приходится увеличивать ε . При этом проверяется, нельзя ли шаг h еще увеличить.

В конце работы блока выдаются на печать следующие величины: $\varepsilon, N, h, z_1(t_0), \dots, z_1(t_N), S$.

Блок 1 работает всякий раз, когда начинает обсчитываться новая допустимая точка x (см. блок 4).

Б л о к 2 (рис.3) вычисляет величины P_i и C_{ij} , а также величину окрестности (10).

Производные $\frac{\partial x_i^0}{\partial x_i^0}$ и $\frac{\partial^2 x_i^0}{\partial x_i^2}$ определяются по формулам численного дифференцирования [6]:

$$\frac{\partial z_i^0}{\partial x_i} = \frac{1}{2\Delta x_i} (z_i^+ - z_i^-),$$

$$\frac{\partial^2 z_i^0}{\partial x_i^2} = \frac{1}{(\Delta x_i)^2} (z_i^+ - 2z_i^0 + z_i^-),$$
(12)

где $z_i^+ = z_i(x_i^0, \dots, x_{i-1}^0, x_i^0 + \Delta x_i, x_{i+1}^0, \dots, x_m^0, t)$,
 $z_i^- = z_i(x_i^0, \dots, x_{i-1}^0, x_i^0 - \Delta x_i, x_{i+1}^0, \dots, x_m^0, t)$.

Счет производных (12) проводится в каждой точке

$$t_k = t_0 + kh, \quad k = 1, \dots, N.$$

Интегралы (8) вычисляются по методу Симпсона с постоянным шагом.

Б л о к 3 (рис.4) после определения локальной допустимой области решает локальную квадратичную задачу и в найденной точке минимума определяет истинное (а не аппроксимационное!) значение функции цели.

Б л о к 4 (рис.5) в случае увеличения целевой функции уменьшает ε для получения более точной аппроксимации и продолжает процесс минимизации, начиная с предыдущей допустимой точки x .

Последовательное уменьшение функции цели продолжается до тех пор, пока величина относительного её уменьшения не сделается меньше текущего значения ε . Затем ε уменьшается, и весь процесс (начиная с выбора h) повторяется.

В результате мы либо по некоторым параметрам получим внутренний минимум, а по остальным выйдем на границу допустимой области $\mathcal{X}$, либо в процессе минимизации относительное уменьшение целевой функции станет меньше заданного числа ε .

По окончании минимизации вычисляются допуски на оптимальные параметры.

Перед остановом машины на печать выдаются:

$$\varepsilon, x, \Delta x, N, h, z_i(t_0), \dots, z_i(t_N), S.$$

Отметим ещё раз, что мы последовательно (по ε) приближаемся к решению: начиная с некоторого значения ε , приходим в

грубую окрестность решения, затем уточняем это решение путем уменьшения окрестности (9) и улучшения аппроксимации. Такой прием позволяет при повышенной точности счета существенно экономить машинное время.

Б л о к 5 в пояснениях не нуждается (см. ПОД SYM в автокодовой программе).

Б л о к 6 - подпрограмма интегрирования (ПОД RUKU) взята с небольшими дополнениями из [9].

Б л о к 7 - нестандартный (подпрограмма счета правых частей системы (I) - ПОД SPZ), объединяется с основной подпрограммой так же, как и ПОД RNZ, с помощью корректировочного оператора ВСТАВИТЬ. Объем ПОД SPZ ≤ 1500 ячеек памяти. В подпрограмме SPZ должно произойти присвоение переменной $z1$ текущего значения выходной функции, переменной $z0$ - значения разности $z_i - z_{i*}$.

Б л о к 8 - нестандартный (ПОД RNZ), рассчитывает начальное значение подынтегральной функции в (3) для $t = t_0$.

В операторе ВВОД описываются:

- начальное значение относительной погрешности ε ;
- её нижняя граница ε_1 ;
- начальный шаг интегрирования h ;
- число оптимизируемых параметров m ;
- векторы ограничений a и b ;
- вектор оптимизируемых параметров x ;
- вектор начальных условий y^0 ;
- N - число точек разбиения интервала T , соответствующее шагу h ;
- число оптимизируемых параметров m_x ;
- порядок системы (I) n_y .

Величины всех остальных параметров системы, не подлежащих оптимизации, описываются числами в ПОД SPZ.

Пример оптимизации параметров усилителя - формирователя является иллюстрацией изложенной выше методики. Выбран упрощенный вариант схемы [2]: случай отсутствия насыщения в транзисторе.

Для удобства счета введены размерные масштабные коэффици-

Перфорируется фактическое количество компонент перечисленных массивов.

енты: $\tau = 10^{-6}$ сек, $z = 100$ ом, $i = 0,01$ а.

Система (I) для усилителя-формирователя с учетом сделанных выше замечаний имеет вид:

$$x_0 = x_1 - x_9,$$

$$x_1 = 4,3 x_6,$$

$$x_2 = \frac{0,6}{3y_3 + 1},$$

$$x_3 = \frac{-0,033 \cdot 5,26 y_3 + 1,7 - y_4}{x_1 + x_2 + 0,033},$$

$$x_5 = 5,26 y_3 - x_2 y_4 + x_3,$$

$$x_6 = 5,26 y_3 - y_4,$$

$$x_7 = -x_5 y_2 + x_3,$$

$$x_9 = \begin{cases} \frac{6,75}{0,3} y_4, & \text{если } 0 \leq y_4 \leq 0,3, \\ 6,75, & \text{если } 0,3 \leq y_4 \leq 1,5, \\ \frac{6,75(2,56 - y_4)}{1,06}, & \text{если } 1,5 \leq y_4 \leq 2,56. \end{cases}$$

$$\begin{aligned} \frac{dy_1}{dt} &= x_4 x_5, & \frac{dy_2}{dt} &= -y_3^2 - 0,363 - 0,25 y_3 + x_3, \\ \frac{dy_3}{dt} &= \frac{4,3 x_6}{x_2}, & \frac{dy_4}{dt} &= 1. \end{aligned}$$

Начальные условия: $y_1(0) = \dots = y_4(0) = 0$.

В качестве оптимизируемых параметров взяты следующие (ср. с [2]): $x_1 = \tau$, $x_2 = r$, $x_3 = L$, $x_4 = \frac{1}{C}$, $x_5 = \mu$.

Ограничения на параметры:

$$\alpha_1 = 1; \quad \alpha_2 = 0,5; \quad \alpha_3 = 3; \quad \alpha_4 = 0,5; \quad \alpha_5 = 0,1;$$

$$\beta_1 = 2; \quad \beta_2 = 1,5; \quad \beta_3 = 7; \quad \beta_4 = 1,5; \quad \beta_5 = 0,6.$$

Проведен счет четырех вариантов, причём исходная точка x^0

выбиралась в разных местах допустимой области X .

Текст автокодовой программы, объединенный с ПОД SPZ и ПОД RNZ для рассматриваемого примера, приведен в приложении.

Результаты счета сведены в табл. I и представлены на рис. 6-9, где цифры у кривых обозначают номера вариантов таблицы:

Таблица I

№ вар- ианта	ε^0	x_1^0	x_2^0	x_3^0	x_4^0	x_5^0	S^0	$\hat{\varepsilon}$	$\hat{x}_1$	$\hat{x}_2$	$\hat{x}_3$	$\hat{x}_4$	$\hat{x}_5$	$\hat{S}$
1	0,32	1,5	1,0	5,0	1,0	0,36	17,6	0,16	2,0	1,5	7,0	0,5	0,15	2,53
2	0,16	1,5	1,0	5,0	1,0	0,36	17,6	0,16	2,0	1,5	6,8	0,5	0,14	2,42
3	0,16	1,1	0,6	3,2	1,4	0,5	53,7	0,04	1,69	1,5	6,68	0,51	0,16	1,75
4	0,16	1,9	1,0	6,8	1,4	0,5	13,1	0,04	1,52	1,5	7,0	0,64	0,15	1,62

(значками $(^0)$ и $(\wedge)$ помечены соответственно исходные и оптимальные кривые). Среднее время счета для одного варианта ~ 4 мин.

Заключение

1. Задача оптимизации параметров динамической системы формулируется как задача минимизации функционала, представляющего квадрат отклонения выходной функции системы от заданной. Выходная функция получается из решения системы дифференциальных уравнений. Допустимой областью является параллелепипед с границами, параллельными координатным гиперплоскостям.

2. Путем линеаризации выходной функции по приращениям параметров задача сводится к последовательности квадратичных программ.

3. В результате процесса оптимизации определяются также допуски на оптимальные значения параметров.

4. Приведена универсальная программа оптимизации, реализующая изложенную методику. Программа написана на языке АКИ для ЭЦМ "Минск-2". Для пользования программой необходимо составить нестандартные блоки счета правых частей системы уравнений и расчета начального значения функционала и задать исход-

ные данные в требуемом порядке.

6. Проведенные расчеты для одной из конкретных схем вычислительной техники показали достаточную эффективность программы.

ЛИТЕРАТУРА

1. В.К.КОРОЛЕВ. Расчет оптимальных параметров импульсного трансформатора.-Вычислительные системы, "Наука", Новосибирск, 1967, вып.27.
2. В.К.КОРОЛЕВ. Оптимизация параметров усилителя-формирователя, там же.
3. Б.М.КАГАН и Т.М.ТЕР-МИКАЭЛЯН. Решение инженерных задач на ЦВМ."Энергия", М.-Л., 1964.
4. Г.ЗОЙТЕНДЕЙК. Методы возможных направлений. ИЛ., М., 1963.
5. И.Г.ПЕТРОВСКИЙ. Лекции по теории обыкновенных дифференциальных уравнений. ГИТТЛ, М., 1952.
6. И.С.БЕРЕЗИН и Н.П.ХИДКОВ. Методы вычислений, т.1, Физматгиз, М., 1962.
7. Б.П.ДЕМИДОВИЧ и И.А.МАРОН. Основы вычислительной математики. Физматгиз, М., 1963.
8. Г.КЮНИ и В.КРЕЛЛЕ. Выпуклое программирование. "Советское Радио", М., 1965.
9. М.Е.НЕМЕНМАН и ДР. Автокод для решения инженерных задач на машине "Минск-2". Минск, 1965.
10. В.И.ПРОХОРОВ и ДР. Основы программирования для ЭЦВМ."Высшая школа". М., 1967.

Поступила в редакцию
19/VI - 1968 г.

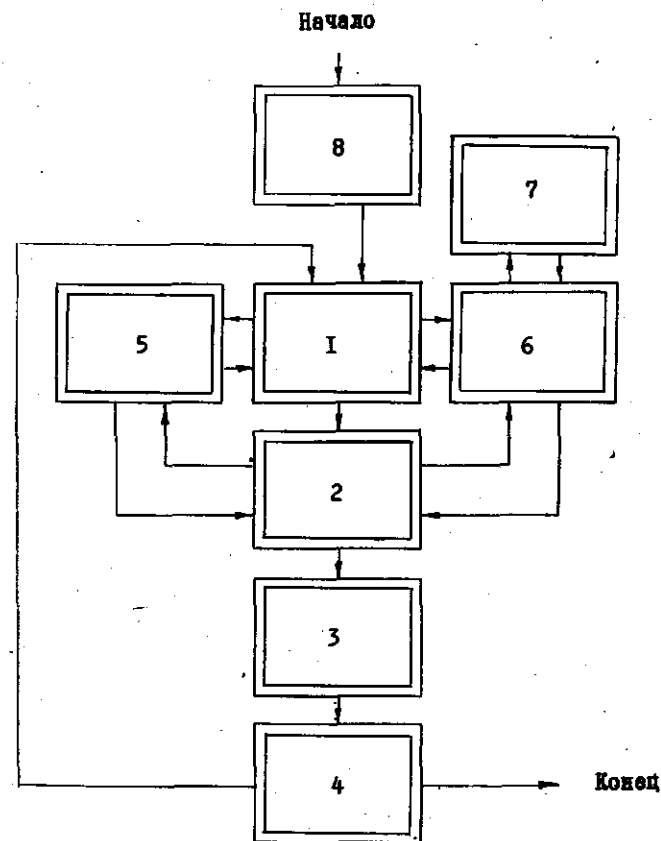


Рис. I

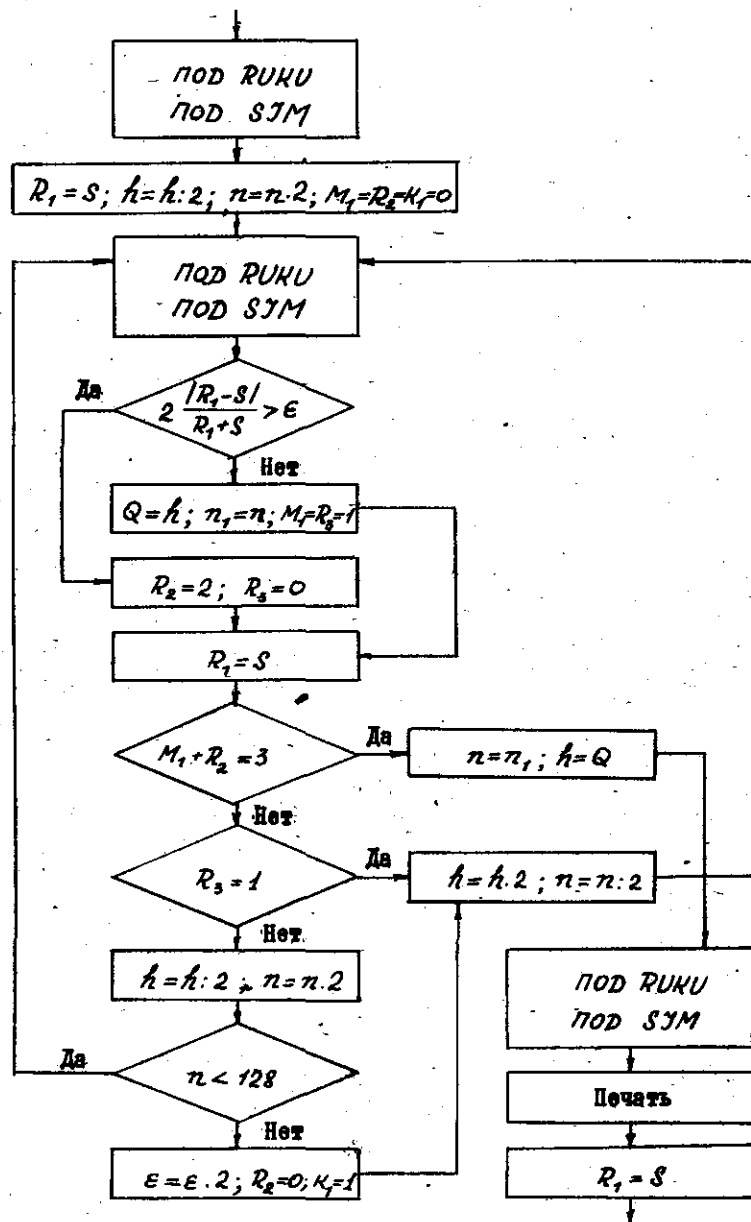


Рис. 2

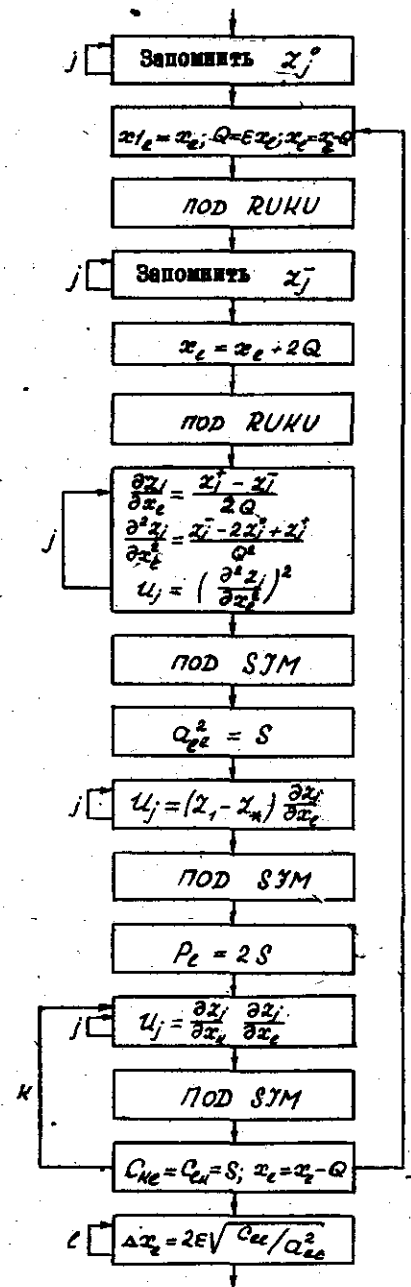


Рис. 3

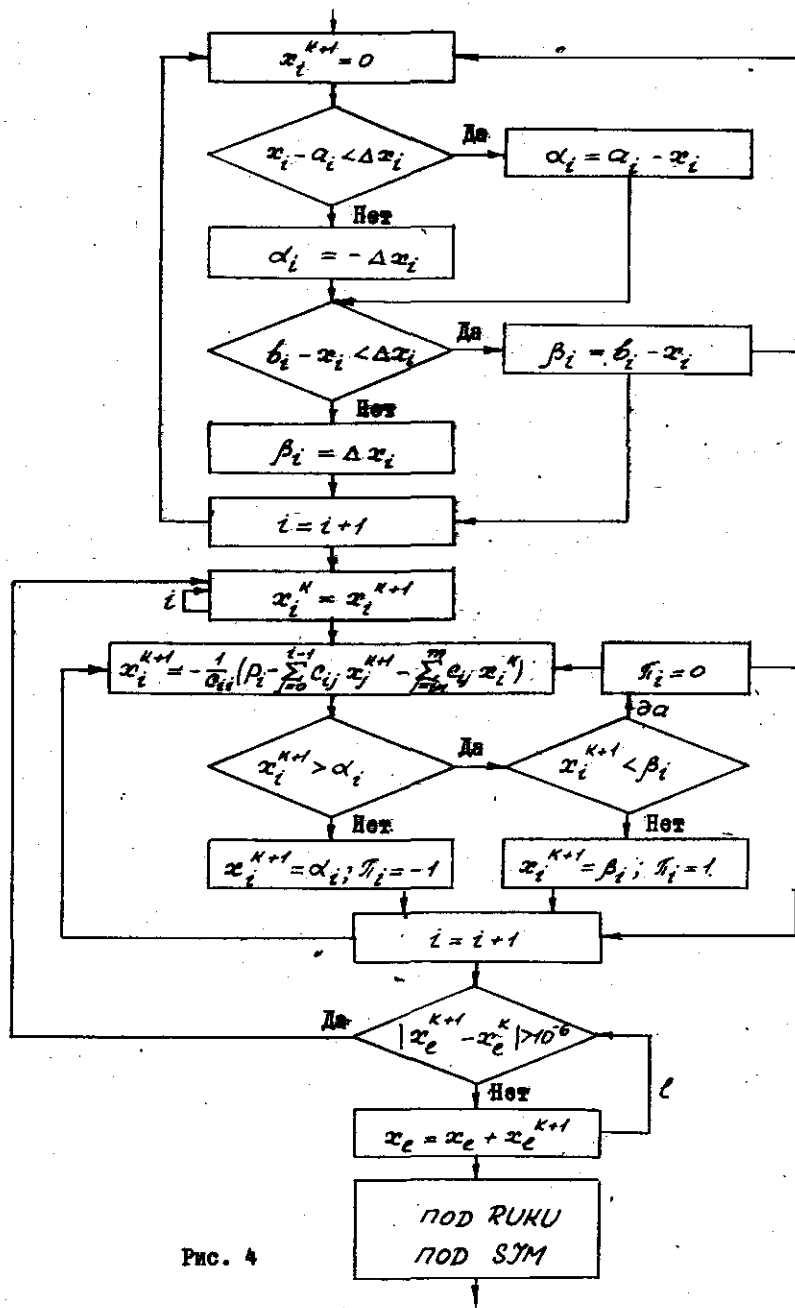


Рис. 4

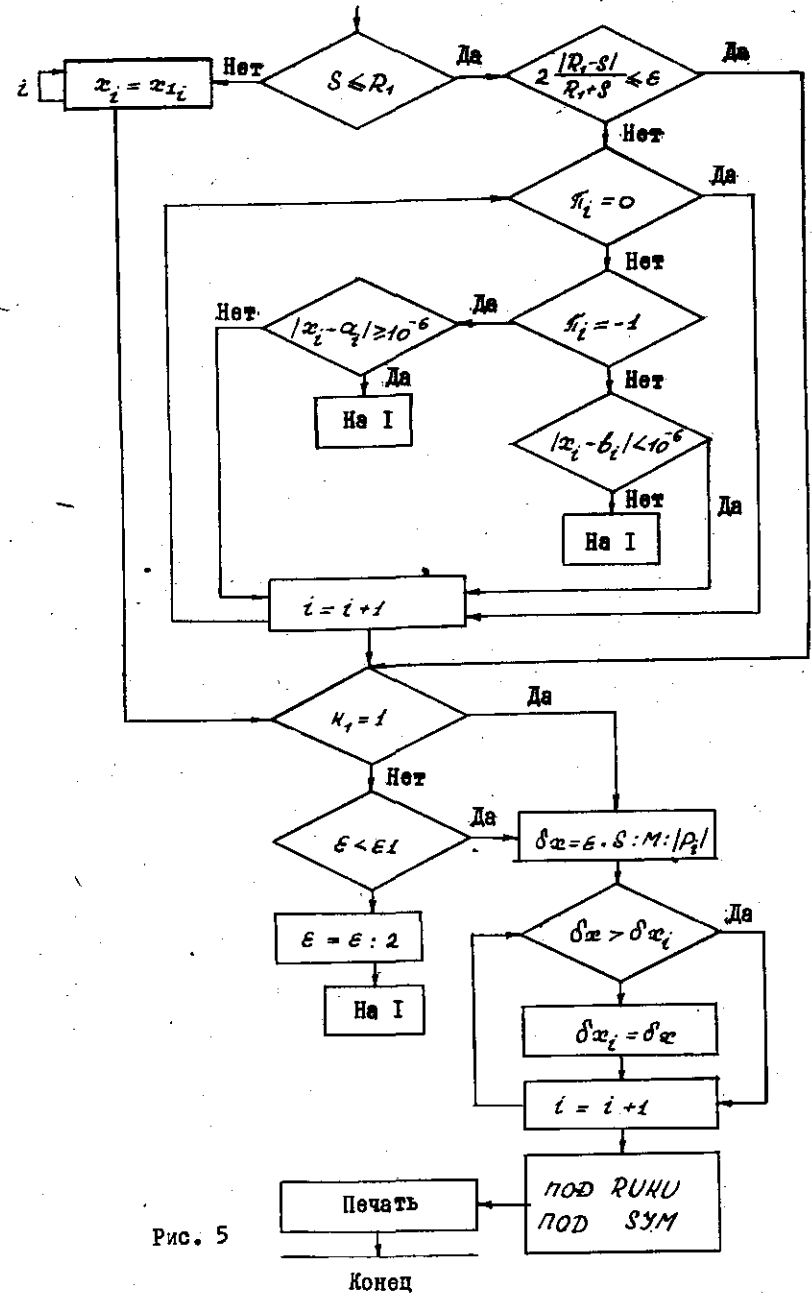


Рис. 5

ЛИСТ 01

01 ПОПАДИСХ.
02 MAC AL(20),C(400 20.20),F(20),P(20),DX(20),I(20),
Y1(20),YB(20) ≡
03 U(64),V(64),W(64),Z(64),DZ(1280 20.64),V1(64)X
04 HAS PL=0.0,5.0,5.1 QL=1.2.2.1 PI(20)=AL/1/ XL(20)
=V/1/ XP(20)≡
05 =V/21/ X1(20)=V/41/ XK(20)=W/1/ XK1(20)=W/21/X
06 1. BBO E, E1, H, M, A(20), B(20), X(20), YN(20): H, MX, NYX
07 BHI 55X
08 BHI 46X
09 BHI 43X
10 2. BCI :N =64 TO 56X
11 BHI R1=8 H=H:2X
12 BHI :N=N.2 M1=0 R2=0 K1=0X
13 3. BHI 46X
14 BHI 43X
15 ECI 2. MOD(R1-8):(R1+8))E TO 4X
16 BHI Q=HX

ЛИСТ 02

01 BHI :N1=N M1=1 R3=1X
02 IEP 5X
03 4. BHI :R2=2 R3=0X
04 5. BHI R1=8X
05 ECI :M1+R2 =3 TO 7X
06 ECI :R3 =1 TO 6X
07 ECI :N =64 TO 56X
08 BHI H=H:2X
09 BHI :N=N.2X
10 6. ECI :N =4 TO 7X
11 BHI H=H.2X
12 КОД 6400 7003 N.-3000 0003 0000.-3077 00Q1 0000.
0000 0000 0101X
13 56. BHI :K1=1X
14 IEP 8X
15 7. BHI :N=N1X
16 BHI H=QX

ЛИСТ 03

01 BHI 46X
02 BHI 43X
03 HAI HA BPM E,X(20),:N,H,Z(64),SX

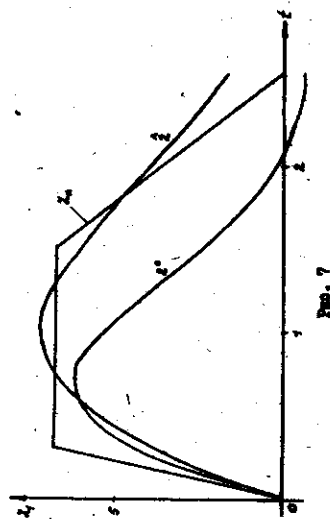


Fig. 7

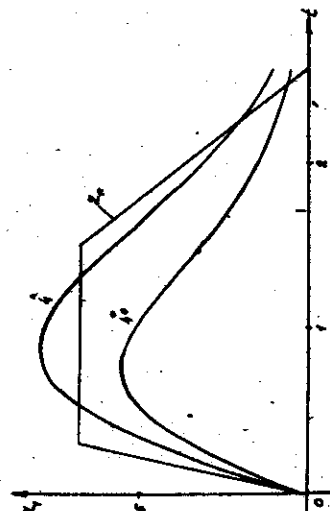


Fig. 9

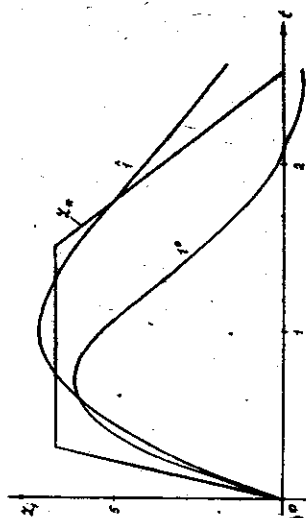


Fig. 6

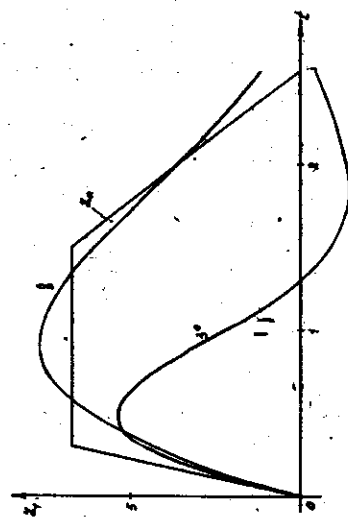


Fig. 8

04 БМЧ R1=8X
05 8.БМЧ :N1=1X
06 9.БМЧ V/J/=Z/J/X
07 ПОВ 9 J=1 (1) NX
08 10.БМЧ X1/L/=X/L/ Q=R1.X/L/ X/L/=X/L/-QX
09 БМЧ 46X
10 11.БМЧ W/J/=Z/J/X
11 ПОВ 11 J=1 (1) NX
12 БМЧ X/L/=X/L/+2.QX
13 БМЧ 46X
14 12.БМЧ DZ/L,J/=(Z/J/-W/J/):2:Q U/J/=((W/J/-2.V/J/+Z/J/)
:Q*2)*2X
15 ПОВ 12 J=1 (1) NX
16 БМЧ 43X

ЛИСТ 04

01 БМЧ AL/L/=8X
02 13.БМЧ U/J/=V1/J/.DZ/L,J/X
03 ПОВ 13 J=1 (1) NX
04 БМЧ 43X
05 БМЧ P/L/=2.SX
06 14.БМЧ U/J/=DZ/K,J/.DZ/L,J/X
07 ПОВ 14 J=1 (1) NX
08 БМЧ 43X
09 БМЧ C/K,I/=C/L,K/=8X
10 ПОВ 14 K=1 (1) N1X
11 БМЧ :N1=N1+1X
12 БМЧ X/L/=X/L/-QX
13 ПОВ 10 L=1 (1) MXX
14 15.БМЧ X=C/K,I/ DX=DX:AL/L/X
15 КОД -3300 0030 0021.-3077 0000 0000X
16 16.БМЧ DX/L/=2.E.DX'(1:2)X

ЛИСТ 05

01 ПЕР 18X
02 17.БМЧ DX/L/= D18X
03 18.ПОВ 15 K=1 (1).L=1 (1) MXX
04 19.БМЧ S=X/I/-A/I/ R=DX/I/ XK1/I/=OX
05 ЕСЛ S (R TO 20X
06 БМЧ XL/I/=-RX
07 ПЕР 21X
08 20.БМЧ XL/I/=-8X
09 21.БМЧ S=B/I/-X/I/X
10 ЕСЛ S (R TO 22X
11 БМЧ XP/I/=RX
12 ПЕР 23X
13 22.БМЧ XP/I/=8X

14 23.ПОВ 19 I=1 (1) MXX
15 24.БМЧ XK/I/=XK1/I/X
16 ПОВ 24 I=1 (1) MXX

ЛИСТ 06

01 БМЧ :N1=0 M1=2X
02 25.БМЧ XK1/I/=-P/I/X
03 ЕСЛ :N1 =0 TO 27X
04 26.БМЧ XK1/I/=XK1/I/-C/I,J/.XK1/J/X
05 ПОВ 26 J=1 (1) N1X
06 27.ЕСЛ :M1-1 =MX TO 29X
07 28.БМЧ XK1/I/=XK1/I/-C/I,J/.XK/J/X
08 ПОВ 28 J=M1 (1) MXX
09 БМЧ XK1/I/=XK1/I/:C/I,K/X
10 29.ЕСЛ XK1/I/)XL/I/ TO 30X
11 БМЧ XK1/I/=XL/I/ PI/I/=-1X
12 ПЕР 32X
13 30.ЕСЛ XK1/I/ (XP/I/ TO 31X
14 БМЧ XK1/I/=XP/I/ PI/I/=1X
15 ПЕР 32X
16 31.БМЧ PI/I/=OX

ЛИСТ 07

01 32.БМЧ :N1=N1+1 M1=M1+1X
02 ПОВ 25 I=1 (1).K=1 (1) MXX
03 33.ЕСЛ MOD(XK1/I/-XK/L/)) 0-6 TO 24X
04 БМЧ X/L/=X/L/+XK1/L/X
05 ПОВ 33 L=1 (1) MXX
06 БМЧ 46X
07 БМЧ 43X
08 ЕСЛ S (=R1 TO 35X
09 34.БМЧ X/I/=X1/I/X
10 ПОВ 34 I=1 (1) MXX
11 ПЕР 39X
12 35.ЕСЛ 2.MOD(R1-S):(R1+S)-(-E TO 39X
13 36.ЕСЛ MOD(PI/I/) (0,5 TO 38X
14 ЕСЛ PI/I/ (-0,5 TO 37X
15 ЕСЛ MOD(X/I/-B/I/) (0-6 TO 38 ИНАЧЕ 2X
16 37.ЕСЛ MOD(X/I/-A/I/))=0-6 TO 2X

ЛИСТ 08

01 38.ПОВ 36 I=1 (1) MXX
02 39.ЕСЛ :K1 =1 TO 40X
03 ЕСЛ E (E1 TO 40X

04 ВМЧ R=E:2X
 05 ПЕР 2X
 06 40. ВМЧ DX=E.S:M DX=DX:MOD(P/I)X
 07 КОД -3300 0051 0052.-3077 0000 0000X
 08 41. ВМЧ DX)=DX/I/ TO 42X
 09 ВМЧ DX/I/=DX
 10 42. ПОВ 40 I=1 (1) MIX
 11 ВМЧ 46X
 12 ВМЧ 43X
 13 НАП НА BPM B,X(20),DX(20),:N,H,Z(64),SX
 14 КОМ X
 15 43. ПОД 81MX
 16 ВМЧ S=80X

ЛИСТ 09

01 ВМЧ :R=M-1X
 02 44. ВМЧ S=S+4.U/J/ S=S+2.U/J+1/X
 03 ПОВ 44 J=1 (2) MX
 04 45. ВМЧ S=S-U/J/X
 05 ПОВ 45 J=M (1).1X
 06 ВМЧ S=S.H:3X
 07 ВМЧ X
 08 46. ПОД 8000X
 09 47. ВМЧ YR/I/=YR/I/ Y1/I/=YR/I/X
 10 ПОВ 47 I=1 (1) MIX
 11 48. ВМЧ Y/I/=YR/I/+F/I/.PI/K/.HX
 12 ПОВ 48 I=1 (1) MIX
 13 ВМЧ 51X
 14 49. ВМЧ Y1/I/=Y1/I/+H:6.F/I/.QI/K/X
 15 ПОВ 49 I=1 (1) MIX
 16 ПОВ 48 K=1 (1) 4X

ЛИСТ 10

01 50. ВМЧ YR/I/=Y1/I/X
 02 ПОВ 50 I=1 (1) MIX
 03 ВМЧ V1/J/=20 U/J/=20'2 2/J/=21X
 04 ПОВ 48 J=1 (1) MX
 05 ВМЧ X
 06 51. ПОД 8P4X
 07 ВМЧ Z2=0,6:(3.Y/3/+1) Z3=(-0,033.5,26.Y/3/+1,7-X/1/)
 : (X/1/+Z2+ ≡
 08 0,033) Z5=5,26.Y/3/-X/2/.X/1/+Z3 Z6=5,26.Y/3/-X/2/ ≡
 09 Z7=X/6/+X/3/.NIP(-X/5/.Y/2/) Z1=4,3,26X
 10 ВМЧ Y/4/)0,3 TO 52X
 11 ВМЧ Z9=6,75.Y/4/:0,5X
 12 ПЕР 54X
 13 52. ВМЧ Y/4/)1,5 TO 53X

14 ВМЧ Z9=6,75X
 15 ПЕР 54X
 16 53. ВМЧ Z9=6,75.(2,56-Y/4/):1,06X

ЛИСТ II

01 54. ВМЧ Z0=Z1-Z9 F/1/=X/4/.Z5 F/2/=4,3.Z6:Z7 F/3/=-Y/3/ /
 2.0,363 ≡
 02 -0,25.Y/3/+Z3 F/4/=1X
 03 ВМЧ X
 04 55. ПОД 8N0X
 05 ВМЧ SO=0X
 06 ВМЧ X
 07 НАЧ 1X
 08