

Головко Леонид Степанович

**Теория погрешностей
Алгоритмы
для реализации в ПЭВМ**

Содержание

Введение	3
1. Теория погрешностей.....	6
1.1. Законы распределения погрешностей.....	6
1.2. Параметры законов распределения погрешностей и их оценки по экспериментальным данным	7
1.3. Энтропийный коэффициент. Типы законов распределения.....	9
1.4. Статистическая устойчивость.....	11
2. Элементы структурного программирования.....	12
3. Алгоритмы обработки измерительной информации.....	16
3.1. Алгоритм “Выборка” (АВ).....	16
3.2. Алгоритм “Среднее” (АС).....	19
3.3. Алгоритм “Средняя квадратичная погрешность” (АСКП).....	22
3.4. Алгоритм “Асимметрия” (АА).....	25
3.5. Алгоритм “Контрэксцесс” (АК).....	28
3.6. Алгоритм “Размах выборки” (АРВ).....	32
3.7. Алгоритм “Энтропийный коэффициент” (АЭК).....	34
3.8. Алгоритм “Закон распределения” (АЗР).....	39
3.9. Алгоритм “Статистическая устойчивость” (АСУ).....	42
4. Закон распределения Пуассона. Счетчики элементарных частиц.....	50
4.1. Алгоритм “Число частиц” (АЧЧ).....	50
4.2. Алгоритм “Закон распределения Пуассона” (АЗП).....	52
5. Алгоритм фильтрации сбоев “Срединное” (АФС).....	55
6. О центрах разработки программного обеспечения для ПЭВМ.....	59
Список литературы	61

Введение

Цель работы обозначена в ее названии. К этому можно добавить, что теория погрешностей изложена в полном соответствии с литературой, приведенной в перечне и только в объеме необходимом для разработки соответствующих алгоритмов. Что касается возможной полезности настоящей работы, то на это отвечает многочисленная литература по проблеме точности средств измерений; попытки в конкретной практике использовать характеристики точности соответствующие теории погрешностей.

Определение характеристик точности в полном соответствии с теорией погрешностей особенно актуально при разработке:

- сложных измерительных приборов с последовательным, ступенчатым преобразованием входного сигнала в выходной;
- информационно-измерительных систем (ИИС);
- систем управления (СУ) разного назначения и степени автономности, поскольку управляющий сигнал рассчитывается по показаниям соответствующих средств измерений.

Наиболее перспективно их создавать по агрегатному принципу, т.е. собирать из отдельно выпускаемых и самостоятельно аттестуемых узлов, агрегатов, приборов. И в этом случае, необходимы такие характеристики точности, которые позволяли бы проводить расчетное суммирование погрешностей. Никакие другие оценки погрешностей в том числе так называемые “максимальные” или “предельные” для этого непригодны. Они годны лишь для грубых, прикидочных расчетов и то для случая суммирования 2-3 отдельных погрешностей [3,6,7,8].

Еще один прием оценки погрешностей заключается в том, что их рассчитывают вводя ничем не обоснованное предположение о том, что вид закона распределения погрешностей точно известен. По 20÷30 отдельным измерениям вычисляют среднюю квадратичную погрешность σ , а затем указывают “максимальную” погрешность $\Delta_m = 3\sigma$ с доверительной вероятностью 0,997 на основании предположения о нормальности закона распределения погрешностей.

Цитата из [3] “... такой прием является “научно” замаскированным обманом вне зависимости от того допускается он сознательно или бессознательно. Дело заключается в том, что реальные законы распределения погрешностей приборов весьма разнообразны и часто далеки от нормального.

Для установления действительного хода кривой распределения на ее краях необходимо проведение испытаний число которых должно быть тем больше, чем больше выбирается значение доверительной вероятности. При малом числе испытаний (20÷30) какие-либо сведения о ходе кривой в районе соответствующем $P_D = 0,995 \div 0,997$, отсутствуют и утверждения о ходе кривой распределения в этом неисследованном районе лишены оснований”.

К выше сказанному можно добавить, что нормальный закон распределения погрешностей (закон Гаусса) в его идеальном виде практически не реализуем в любых конкретных условиях вследствие его основного свойства – максимальной энтропии [3,7] т.е. в случае реальных законов распределения могут иметь место только приближения какой-то степени. Это подтверждается и центральной предельной теоремой статистики [1,10], которая утверждает, что нормальный закон распределения погрешностей может быть получен, когда количество суммируемых случайных переменных стремится к бесконечности.

В духе нашего времени тот факт, что если предполагается стыковка средств измерения с ЭВМ (например, системы управления) у разработчиков как проектной, так и эксплуатационной документации возникает желание использовать вычислительные возможности ЭВМ, пытаясь сформулировать характеристики точности в соответствии с

теорией погрешностей. При этом такие формулировки выглядят неполными, недостаточными, а то и абсурдными.

В соответствии с теорией погрешностей характеристиками точности являются средняя квадратичная погрешность, доверительный интервал и соответствующая ему доверительная вероятность. В каких-то случаях можно ограничиться последними двумя (например, если нет необходимости суммирования погрешностей).
Фрагменты из практики разработки СУ.

В проектной документации (например, техническое задание) обозначают численное значение погрешности и указывают, что оно должно соответствовать 3σ и больше ничего. В такой формулировке можно только догадываться о том, что требуют, чтобы закон распределения погрешностей разрабатываемого прибора был нормальным, а указанная погрешность соответствовала бы доверительной вероятности 0,997, поскольку употреблен коэффициент 3. Это все тот же подход так называемых ‘запасов’, который вообще не имеет смысла, поскольку закон распределения погрешностей разрабатываемого прибора может не соответствовать нормальному, а численное значение погрешности может соответствовать доверительной вероятности 0,997 при коэффициентах меньше 3. Разработчик прибора при этом вполне бессознательно пытается обеспечить численное значение погрешности вне всякой связи с теорией погрешностей но с ‘запасом’.

Еще пример, вполне характерный для современного понимания возможностей вычислительной техники.

Разработчик измерительного прибора предназначенного для СУ теперь уже в эксплуатационной документации (паспорт, ТУ) непонятно зачем, приводит алгоритм, как будто разработанный по методу структурного программирования, для вычисления средней квадратичной погрешности. Ни по структуре, ни по физическому содержанию он ни за что непригоден и там ему не место. Но, что самое замечательное, так это то, что декларируемое численное значение точности никакого отношения к средней квадратичной погрешности не имеет.

Для того чтобы использовать характеристики точности соответствующие теории погрешностей, необходимы технические решения, обеспечивающие получения выборок достаточного объема и разнообразия значений отдельных измерений. По отношению к СУ и ИИС это может означать возможность оперативного контроля этих характеристик в условиях эксплуатации, что существенно повышает надежность измерительной информации и выполнения задач в целом. ‘Шум’ измерительного прибора формируется всеми его деталями и узлами и его характеристики должны оставаться постоянными в течение всего срока эксплуатации в пределах погрешности зависящей от объема используемой выборки.

Чрезвычайно высокая трудоемкость получения ‘вручную’ достаточно больших объемов выборок и их обработки не позволяют эффективно использовать параметры теории погрешностей с целью контроля качества продукции при ее массовом или серийном производстве (например, транзисторов разного назначения). Хотя такие попытки были [1]. Тем не менее, оценить объем контролируемой выборки, частоту контроля, статистическую устойчивость технологического процесса невозможно без использования теории погрешностей. Иначе количество регистрируемого брака будет весьма проблематичным и, следовательно, эффективность производства в целом.

В настоящее время состыковать ПЭВМ с приборами, измеряющими отклонение от номинала, технической проблемы нет. И таким образом, почти полностью автоматизировать контроль качества технологического процесса в целом. Оператору остается обеспечивать электрический контакт в приемном устройстве. Все остальное обеспечит ПЭВМ – накопление выборки нужного объема, ее обработку, документирование и выдачу на экран монитора результатов.

Выше обозначенные проблемы невозможно решить одномоментно - необходимо накопление опыта, где на первом месте разработка соответствующих алгоритмов и

программ, без чего невозможен технический прогресс в каждом конкретном случае и в целом.

В настоящей работе алгоритмы для вычисления параметров теории погрешностей разработаны с использованием методики структурного программирования сверху вниз [10, 11], доступной для освоения любым специалистом своей области знания и не требующей англоязычной терминологии. В качестве исходных данных для разработки алгоритмов и соответствующих программ служат словесное (словесно-формульное) описание задачи или циклограмма управления с комментариями (последовательность выдачи команд, анализ и обработка результатов в случае алгоритмов для СУ). Наиболее эффективной программа будет в том случае, если программист будет иметь достаточно полное представление о задаче во всех ее нюансах, а соответствующий специалист о реализуемом вычислительном процессе, чтобы быть убежденным, что его задача решается правильно во всех возможных вариантах. Таким образом, среди прочих проблем, имеет место проблема взаимопонимания программиста и специалистов разных областей знания. Опыт показывает, что чем сложнее задачи, тем больше возникает трудностей в таком взаимопонимании. В [10] утверждается: “Мир программирования усеян останками программ, которые когда-то считались готовым продуктом, но впоследствии оказались неправильными, неэффективными, непонятными или непригодными по какой-то другой причине. ЭВМ достаточно сложный инструмент и требуется время и усилия для того, чтобы научиться им пользоваться. Не так легко овладеть ЭВМ, как мощным инструментом для решения задач, особенно математических.” В США подсчитали, что от 50 до 100% времени программист тратит на устранение ошибок и реализацию недостающих вариантов задач [10]. Поиски таких методик разработки алгоритмов, которые уменьшали бы вероятность ошибок в программах, упрощали бы их понимание и облегчали модификацию привели к тому, что Бомом и Якопини была предложена, так называемая, методика структурного программирования сверху-вниз, которая завоевала наибольшую популярность в США [10, 11]. Под структурным программированием понимают процесс разработки алгоритмов с помощью структурных блок-схем. Соответствующие блок-схемы отражают управление вычислительным процессом и таким образом, устанавливают нужную последовательность вычислительных операций со всеми ее возможными вариантами в процессе решения какой-либо задачи в ЭВМ и поэтому алгоритмы приобретают, в некотором смысле, свойство правильности, встроенное в них шаг за шагом и, следовательно в них меньше ошибок, а их правильность доказывается легче. Ясно, однако, что такого эффекта можно достичь только тогда, когда в блоках будут отражены не общие пожелания, а конкретные действия по управлению вычислительным процессом и указания о вычислительных операциях должны подразумевать использование хорошо известных стандартных подпрограмм.

Структурные блок-схемы алгоритмов могут разрабатывать специалисты своей области знания по правилам общепонятным как любому программисту, так и соответствующему специалисту. Эти блок-схемы и могут быть предметом их взаимодействия и наилучшего взаимопонимания. Вполне закономерным выглядит тот факт, что именно физик Питер Менсфилд разработал алгоритмы для магниторезонансного томографа в результате чего стал одним из лауреатов Нобелевской премии в области биологии и медицины за 2003 год. В соответствующем разделе настоящей работы изложены формализованные правила для разработки структурных блок-схем алгоритмов в полном объеме необходимом для практического использования.

Хорошей иллюстрацией применения методики структурного программирования сверху-вниз является разработка алгоритмов для вычисления параметров теории погрешностей. Основной особенностью вычисления обозначенных параметров является огромное количество вычислительных операций, даже для выборок относительно небольших объемов. Тем не менее, построив вычислительный процесс циклически, посредством накопления результата можно сделать алгоритмы компактными, простыми

для понимания, реализации их в виде программных модулей, тестирования и отладки. Объем выборок при этом практически может быть любым.

1. Теория погрешностей

1.1. Законы распределения погрешностей.

Как бы тщательно мы не измеряли какую-либо физическую величину, мы не можем рассчитывать на получение ее истинного значения. Причиной тому является всегда присутствующие случайные погрешности – неопределенные по своей величине и в появлении разных значений которых нам не удастся установить какой-либо закономерности. Они определяются сложной совокупностью причин, трудно поддающихся анализу. Присутствие случайных погрешностей может быть обнаружено при повторных измерениях в виде некоторого разброса результатов. В подавляющем большинстве случаев процесс появления случайных погрешностей есть стационарный случайный процесс. Поэтому разнообразие величин случайных погрешностей характеризуется указанием закона распределения их вероятностей или указанием параметров этого закона. Все выше сказанное в полной мере относится и к технологическим процессам массового и серийного производства (например, транзисторов разного назначения и т.д.). Установить экономическую эффективность такого технологического процесса, не исследовав соответствующий закон распределения, невозможно. Количество брака будет непредсказуемым и не замечаемым, поскольку ни объем контролируемых выборок, ни частоту контроля обосновать нельзя.

Если попробовать представить себе бесконечный процесс повторных измерений в результате чего должен быть получен бесконечный ряд значений отдельных измерений, то такой ряд называют генеральной совокупностью. Однако ясно, что мы всегда имеем дело с ограниченным количеством значений отдельных измерений, которое и будет составлять выборку объема n_0 ,

$$x_1, x_2, x_3, \dots, x_{n_0-1}, x_{n_0}.$$

Разность наибольшего значения отдельного измерения x_{\max} и наименьшего x_{\min} называют размахом выборки $R = x_{\max} - x_{\min}$.

Диаграмму, которая наглядно показала бы как часто получаются те или иные значения называют гистограммой. Чтобы построить гистограмму нужно разбить размах выборки R на равные интервалы d и подсчитать сколько значений отдельных измерений n_i попадает в каждый интервал. Для этих интервалов можно вычислить долю n_i/n_0 полного числа значений отдельных измерений n_0 приходящуюся на каждый интервал. Разделив эту долю на длину интервала d т.е. n_i/n_0d получим так называемую оценку плотности вероятности p_i [3] соответствующего интервала рис. 1.1.

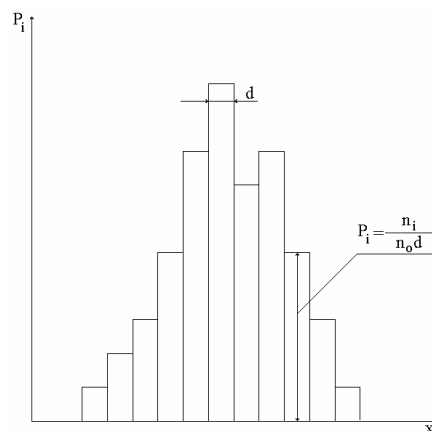


Рис. 1.1.

Если теперь представить себе выборку очень большого объема, а ширину интервала d очень малой (при условии, что измерительный прибор обладает достаточной чувствительностью) и все же в каждом интервале d будет достаточно значений отдельных измерений n_i то должна получиться гладкая кривая $p(x)$ называемая функцией плотности распределения вероятностей [3] и которая может иметь вид изображенный на рис. 1.2, где X в идеальном случае истинное значение измеряемой величины, а σ некий параметр этого распределения.

Исходя из самого способа построения функции плотности распределения вероятностей (т.е. по определению) ясно, что

$$\int_{-\infty}^{+\infty} p(x)dx = 1.$$

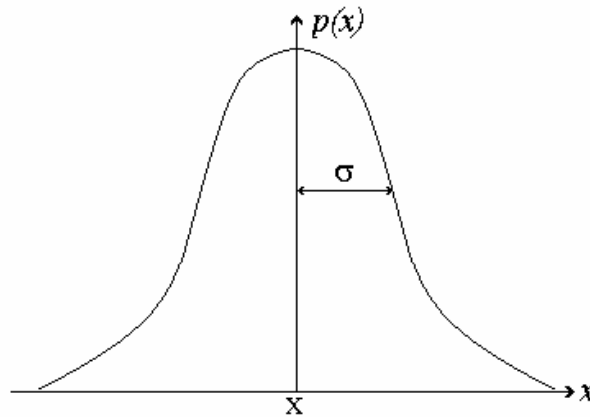


Рис. 1.2.

Смысл функции распределения $p(x)$ состоит в том, что произведение $p(x)dx$ есть вероятность того, что случайно выбранное значение отдельного измерения окажется в интервале от x до $x + dx$. Чаще всего, для краткости, функцию $p(x)$ называют законом распределения погрешностей или просто законом распределения.

1.2. Параметры законов распределения погрешностей и их оценки по экспериментальным данным.

Математическое ожидание определяется как

$$X = \int_{-\infty}^{+\infty} x \cdot p(x)dx.$$

Оценкой математического ожидания есть среднее арифметическое (или просто среднее) выборки объема n_0 [1,3].

$$\bar{x} = \frac{\sum_{i=1}^{n_0} x_i}{n_0},$$

где x_i значения отдельных измерений.

Дисперсия или мощность рассеяния значений x_i вокруг X определяется [1, 3] как

$$D = \int_{-\infty}^{+\infty} (x - X)^2 p(x)dx.$$

Корень квадратный из дисперсии это так называемая средняя квадратичная погрешность [1, 3, 5, 6] т.е. $\sigma = \sqrt{D}$.

σ служит мерой ширины закона распределения (см. рис. 3) и имеет ту же размерность что и измеряемая величина.

Оценка средней квадратичной погрешности вычисляется [1, 3] как

$$\sigma(x_i) = \left[\frac{\sum_{i=1}^{n_0} (x_i - \bar{x})^2}{n_0 - 1} \right]^{1/2}.$$

При точных измерениях $\sigma(x_i)$ мала, а при грубых будет иметь место большой разброс значений отдельных измерений и $\sigma(x_i)$ будет велика.

Асимметрия (т.е. скошенность закона распределения) определяется [1, 3] как

$$\alpha = \frac{1}{\sigma^3} \int_{-\infty}^{+\infty} (x - X)^3 p(x) dx.$$

Оценка асимметрии по экспериментальным данным вычисляется как

$$a(x_i) = \frac{1}{\sigma^3(x_i)} \frac{\sum_{i=1}^{n_0} (x_i - \bar{x})^3}{n_0}.$$

Для симметричных относительно математического ожидания законов распределения $a = 0$.

Эксцесс (или острровершинность) закона распределения определяется как [1, 3]

$$\varepsilon = \frac{1}{\sigma^4} \int_{-\infty}^{+\infty} (x - X)^4 p(x) dx.$$

У различных законов распределения эксцесс изменяется от 1 до ∞ . Для классификации законов по их форме удобнее пользоваться величиной $\varkappa = 1/\sqrt{\varepsilon}$, обратной корню квадратному из эксцесса и изменяющемуся для любых законов распределения от 0 до 1. Эту величину называют контрэксцессом [3].

Оценка эксцесса по выборке объема n_0 может быть вычислена как

$$\varepsilon(x_i) = \frac{1}{\sigma^4(x_i)} \frac{\sum_{i=1}^{n_0} (x_i - \bar{x})^4}{n_0}.$$

Обозначив $\frac{1}{n_0} \cdot \sum_{i=1}^{n_0} (x_i - \bar{x})^4 = \mu(x_i)$ можно переписать

$$\varepsilon(x_i) = \mu(x_i) / \sigma^4(x_i).$$

$\mu(x_i)$ – это оценка так называемого четвертого центрального момента случайной величины [3, 6].

С учетом поправки зависящей от объема выборки [3]

$$\bar{\mu}(x_i) = \frac{n_0(n_0^2 - 2n_0 + 3)}{(n_0 - 1)(n_0 - 2)(n_0 - 3)} \mu(x_i) - \frac{3(n_0 - 1)(2n_0 - 3)}{n_0(n_0 - 2)(n_0 - 3)} \sigma^2(x_i),$$

$$\varepsilon(x_i) = \bar{\mu}(x_i) / \sigma^4(x_i).$$

Таким образом, оценка контрэксцесса может быть вычислена как

$$\varkappa(x_i) = 1/\sqrt{\varepsilon(x_i)} = \sigma^2(x_i) / \sqrt{\bar{\mu}(x_i)}.$$

1.3. Энтропийный коэффициент. Типы законов распределения.

В фундаментальной науке бытует выражение: энтропия - мера беспорядка.

Обусловленное дезинформационным действием помех разнообразие значений отдельных измерений принадлежащих одной и той же генеральной совокупности, также позволяет говорить о степени неупорядочности этих значений и энтропии законов распределения [3, 7].

Доказано [6, 7], что на точность измерений влияет не только мощность помехи (т.е. дисперсия $D = \sigma^2$), но и тип закона распределения. Другими словами, если $\sigma_1 = \sigma_2$ для двух разных законов распределения, то это еще не говорит об одинаковой точности измерений.

Типы законов распределения можно идентифицировать по двум параметрам - энтропийному коэффициенту K_s и контрэкссессу α [3, 6, 7]. Энтропийный коэффициент определяется как отношение энтропийного значения погрешности Δ_s к средней квадратичной погрешности σ т.е. $K_s = \Delta_s / \sigma$. Энтропийное значение погрешности может быть вычислено по экспериментальным данным [3,7].

$$\Delta_s(x_i) = \frac{dn_0}{2} 10^{-\frac{1}{n_0} \sum_{i=1}^{m_0} n_i \lg n_i},$$

где: d – длина интервала на которые разбит размах выборки R .

n_0 – объем используемой выборки.

m_0 – количество интервалов длиной d на которые разбит размах выборки.

n_i – количество значений отдельных измерений попадающих в i -й интервал длиной d .

С учетом поправки на смещение [3] от возможного недостаточно большого числа n_i , попадающих в каждый из интервалов длиной d $\Delta_s(x_i)$ должна быть умножена на

$$\Pi_1 = \left(1 - \frac{m_0}{2n_0} \right),$$

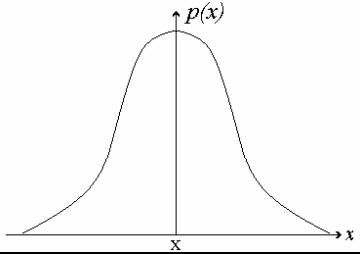
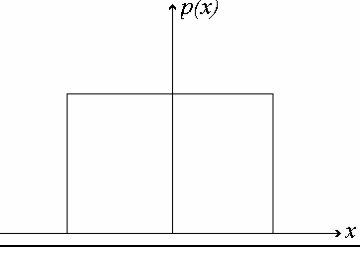
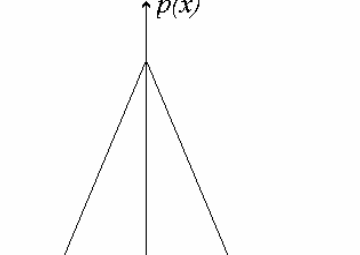
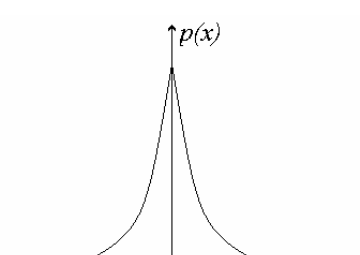
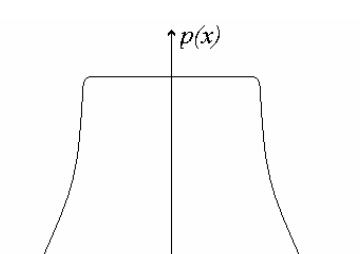
т.е. $\Delta_s^0(x_i) = \Pi_1 \Delta_s(x_i)$. Доказано, что погрешность в $\Delta_s(x_i)$, зависящая от объема используемой выборки n_0 уменьшается с ростом n_0 заметно быстрее, чем соответствующая погрешность в $\sigma(x_i)$ [3].

Перечень законов распределения, которыми можно аппроксимировать реальные законы, приведен в табл. 1 [3, 9].

Кроме того, этот перечень не является исчерпывающим, но, например, с точки зрения результирующей погрешности средств измерений наиболее приемлемый. Из всех возможных законов распределения особое место занимает нормальный (или закон Гаусса), отличающийся тем, что имеет максимальную энтропию, что позволяет аппроксимировать им любые другие, и что тем не менее не всегда оправдано. Кроме того распределение средних \bar{x} асимптотически нормально вне зависимости от вида исходного закона распределения. Считается [3, 7], что уже при объеме выборок $n_0 = 100$ и больше \bar{x}

распределено нормально а $\sigma(\bar{x}) = \frac{\sigma}{\sqrt{n_0}}$.

Таблица 1

Наименование закона распределения $p(x)$	Энтропийный коэффициент K_s	Контрэкссесс α	Вид закона
Нормальный или закон Гаусса	2.066	0.577	
Равномерный	1.73	0.74	
Треугольный	2.02	0.646	
Острове́ршинные	1.35÷1.695	0.199÷0.408	
Трапецеидальные	1,73÷2,02	0,646÷0,74	

1.4. Статистическая устойчивость.

Измерительную информацию можно считать статистически устойчивой, если под воздействием каких-либо факторов не деформируется закон распределения соответствующей генеральной совокупности, в чем можно убедиться контролируя стабильность средней квадратичной погрешности и поведение периферийных областей (краев) этого закона. Контролируя среднюю квадратичную погрешность следует учитывать погрешность возникающую при ее вычислении и зависящую от объема используемой выборки n_0 . Если эту погрешность обозначить Δ , то относительная погрешность в случае нормального закона распределения определяется как $\delta = \Delta / \sigma(x_i) = [1/(2n-1)]^{1/2}$ [5].

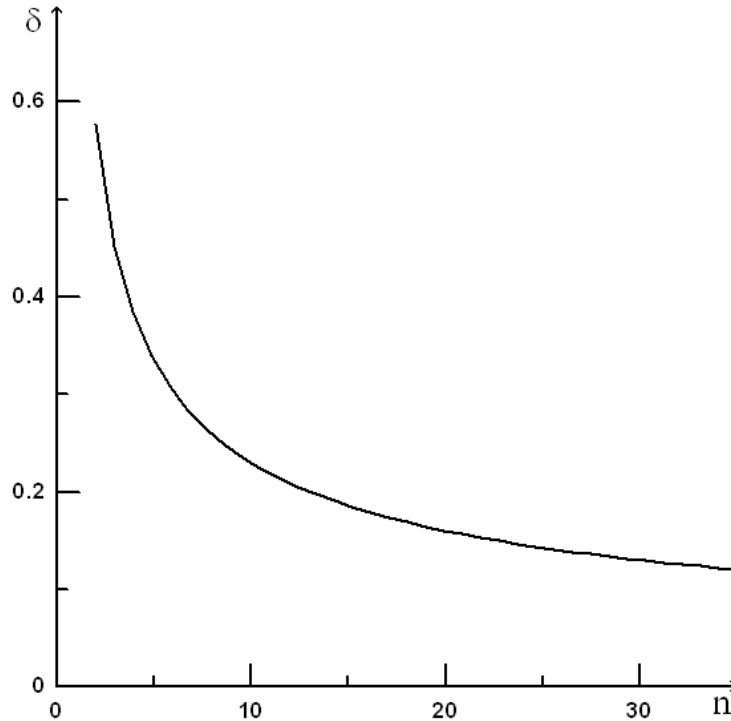


Рис. 1.3.

Из рис. 1.3. видно, что при объемах выборок $n \gg 30$ δ убывает быстро и очень медленно при $n > 30$. Вследствие максимальности энтропии нормального закона распределения можно предположить, что для симметричных законов δ будет меньше при одинаковых объемах выборок.

Имея ввиду достаточно большие объемы выборок можно считать, что среднее \bar{x} достаточно близко к математическому ожиданию т.е. $\bar{x} \approx X$. В случае симметричных законов распределения значения отдельных измерений группируются вокруг \bar{x} , поэтому можно выяснить зависимость вероятности попадания этих значений в интервалы $\bar{x} \pm \Delta x$. Δx здесь удаленность от \bar{x} . Для универсальности можно положить $\Delta x = K\sigma(x_i)$ (т.е. сделать $\sigma(x_i)$ мерой удаленности по оси x от среднего \bar{x}) и принять, что $\Delta x = R/2$, где R размах выборки, тогда $K_{\max} = R/2\sigma(x_i)$. Ясно, что имеют смысл только значения $K < K_{\max}$.

Интервал $\bar{x} \pm K\sigma(x_i)$ называют доверительным, а вероятность P_d с которой x_i попадают в этот интервал, доверительной [2, 8]. Эмпирически P_d соответствует доле x_i попадающих в интервал $\bar{x} \pm K\sigma(x_i)$ т.е. $\Delta n / n_0$. Для нормального закона распределения зависимость P_d от K имеет вид рис. 1.4 [5].

Государственные стандарты регламентируют два значения доверительной вероятности $P_d=0,95$ для большинства областей измерительной техники а для особо ответственных $P_d=0,997$ [8, 9] .

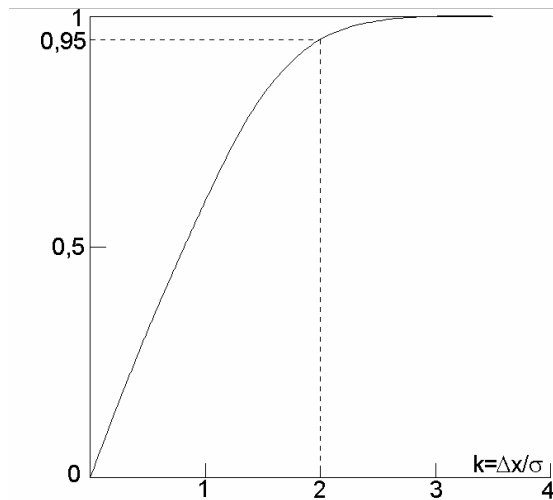


Рис. 1.4.

В случае нормального закона распределения $P_d=0,95$ при $K=2$ и за пределами интервала $\bar{x} \pm 2\sigma(x_i)$ в среднем оказывается одно на 22 отдельных измерений; $P_d=0,997$ при $K=3$ и за пределами интервала $\bar{x} \pm 3\sigma(x_i)$ в среднем оказывается одно на ~400 отдельных измерений [3, 5, 6]. В случае других типов законов распределения коэффициенты будут другими [9]. Измерительная информация будет статистически устойчивой, если количество значений отдельных измерений оказывающихся за пределами интервала $\bar{x} \pm K\sigma(x_i)$ будет стабильным.

2. Элементы структурного программирования

Как известно, алгоритм – это последовательность действий выполнение которой должно приводить к решению поставленной задачи.

Лучшей методикой по разработке алгоритмов будет та, использование которой уменьшает вероятность ошибок в программах, упрощает их понимание и облегчает возможные модификации.

Считается [10], что методика структурного программирования сверху–вниз отвечает вышеперечисленным требованиям и является самой популярной в США.

Любая методика по разработке алгоритмов – это набор общепринятых, формализованных элементов, правил, с использованием которых можно отразить последовательность действий в ЭВМ. Под структурным программированием понимается процесс разработки алгоритмов с помощью структурных блок-схем, которые и отражают эту последовательность событий.

На рис. 2.1. представлены формализованные элементы с использованием которых и строятся блок-схемы.

Содержание блока на рис. 2.1.а $A := f$ отражает так называемую операцию присвоения переменному параметру A значения f , которое может быть результатом каких-либо вычислений, переприсвоением с целью запоминания, командами управления вычислительным процессом или исполнительным органом системы управления. В последних двух случаях используются $0 \vee 1$. Ноль отсутствие команды; единица требования ее выполнения.

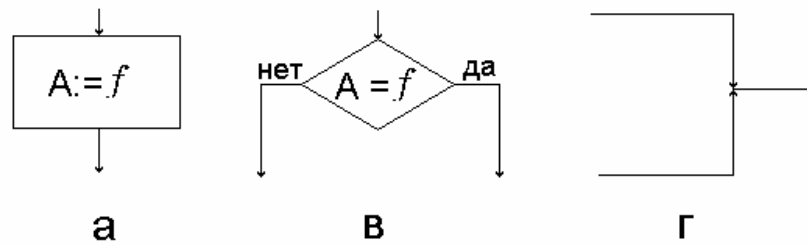


Рис. 2.1.

Блок на рис. 2.1.в отражает то, что можно назвать опросом или анализом в результате которого будет принято решение направить вычислительный процесс по ветви “да” или “нет”. Сравнение A и f может быть не только по $A = f$ но и по $A < f$ и $A \geq f$ т.е. всего возможного вида сравнений A с f , в том числе и вида $f_1 \leq A \leq f_2$. Такого вида процедуры могут быть реализованы в ЭВМ с использованием любого языка программирования.

Возможные обход, возврат, повторение в вычислительном процессе отражаются элементом на рис. 2.1.г. Решения о подобного рода действиях могут возникнуть после блока анализа рис. 2.1.в.

Предполагается [10], что структурная блок-схема может быть выражена как композиция из четырех элементарных блок-схем изображенных на рис. 2.2.

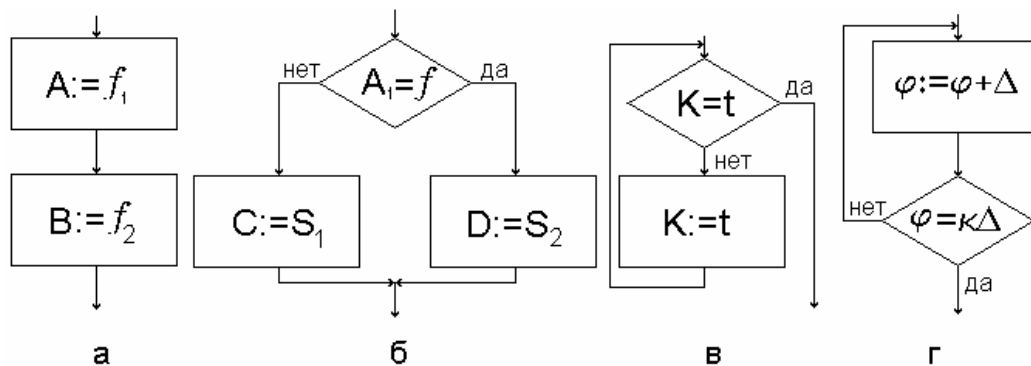


Рис. 2.2.

Рис. 2.2.а в виде двух блоков отражает две операции присвоения, последовательность которых определяется логикой решаемой задачи (или достижения поставленной цели). Эти две операции могут быть записаны и в одном блоке, но в той же последовательности.

На рис. 2.2.б блок анализа определяет какую из операций присвоения произвести по ветви “да” или по ветви “нет”.

На рис. 2.2.в блок анализа определяет нужен ли возврат к повторному анализу по ветви “нет” с блоком операции присвоения или не нужен и тогда вычислительный процесс продолжается по ветви “да”.

На рис. 2.2.г после блока с операцией присвоения следует блок анализа, который определяет нужен ли повтор операции присвоения по ветви “нет” (с последующим повторным анализом) или не нужен и тогда вычислительный процесс продолжается по ветви “да”.

Стоит привести еще одну элементарную структуру, которую с некоторой долей условности можно назвать “счетчиком событий”.

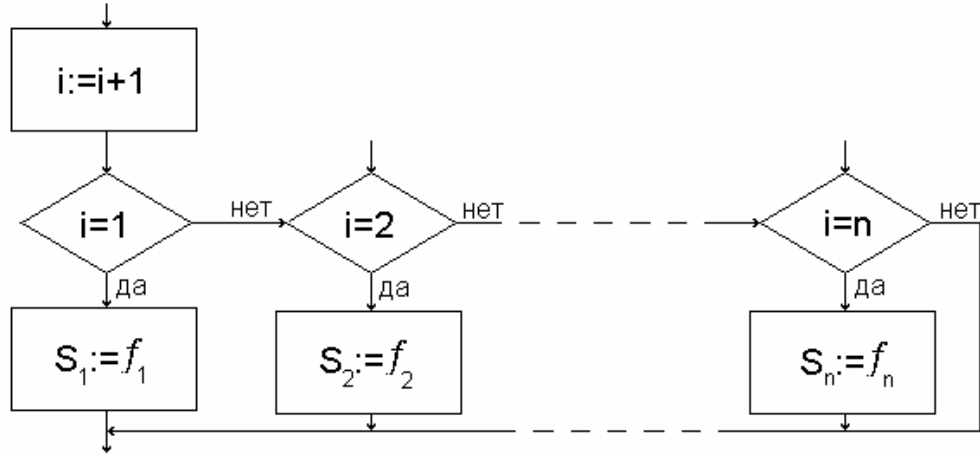


Рис. 2.3.

На рис. 2.3. операция присвоения $i := i + 1$ означает прибавление единицы при каждом повторном заходе на этот блок присвоения. При каждом значении i от единицы до n блоки анализа определяют соответствующие операции присвоения от $S_1 := f_1$ до $S_n := f_n$.

Структурное программирование допускает большее разнообразие структур, чем те которые изображены на рис. 2.2, а само построения блок-схем не обязательно оказывается простейшим. Любая блок-схема разработанная с использованием выше упомянутых элементов и правил должна иметь один вход и один выход. Алгоритм (для примера очень простой) может выглядеть следующим образом рис. 2.4.

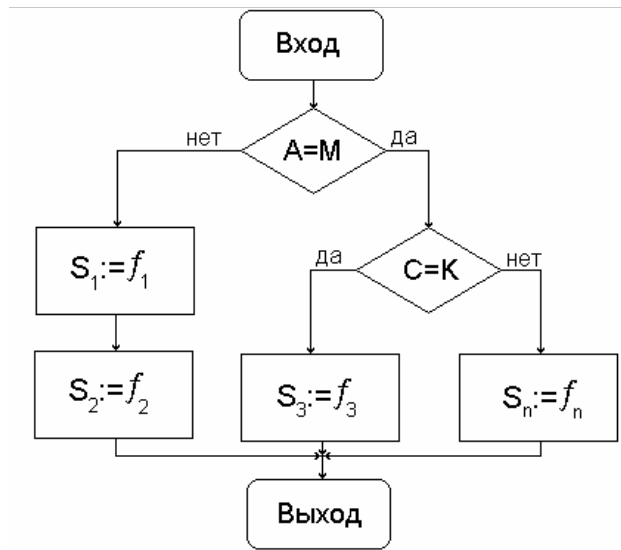


Рис. 2.4

Никакая блок-схема абсолютно ничего не значит без общепринятого, формализованного описания, используемых параметров. В блок-схеме могут использоваться как переменные так и постоянные в конкретном вычислительном процессе параметры. Формализация описания параметров наилучшим образом укладывается в таблицы переменных и постоянных параметров для каждого алгоритма с указанием их характеристик, связанных с принципами реализации цифровых значений этих параметров в ЭВМ и организацией вычислительного процесса. Такие таблицы должны содержать следующие колонки:

1. Наименование параметра – наиболее близкое к его содержанию (назначению).
2. Идентификатор (обозначение), используемое в блок-схеме.
3. Цифровой диапазон возможного изменения (от и до).
4. Дискретность представления параметра в ЭВМ. (т.е. точность их реализации в ЭВМ; пример: 0,123 – дискретность 10^{-3})
5. Где формируются параметры (т.е. указать алгоритмы в которых с этими параметрами производятся соответствующие операции присвоения).
6. Где используются параметры (т.е. указать алгоритмы где производится или опрос их значений, или операция переприсвоения их значения другому параметру или с целью каких-то вычислений).
7. Начальная установка параметров, т.е. задаваемые им значения до начала вычислительного процесса.

Таблица переменных параметров может выглядеть следующим образом:

№ п/п	Наименование параметра	Идентификатор	Диапазон изменения	Дискретность	Где формируется	Где используется	Начальная установка	Примечания

Таблица постоянных параметров должна содержать колонку значения параметра или диапазона коррекций вместо “Диапазон изменения”, а также должны отсутствовать колонки “Где формируется” и “Начальная установка”.

Чтобы облегчить проверку программы на правильность функционирования используют так называемый модульный принцип разработки алгоритмов и соответствующих программных модулей (ПМ) (т.е. задачу разбивают на более-менее логически завершенные части). Программа должна быть проверена для широкого спектра допустимых входных данных по всем ветвям блок-схемы. Если обратиться к рис. 2.4. то в алгоритме можно насчитать всего три ветви, правильность функционирования которых можно проверить задавая на вход соответствующие данные и контролируя выходные. Что касается проверки решения какой-либо задачи в целом, то соответствующую модель может предложить только специалист соответствующей области знания.

Исходными данными для разработки алгоритмов могут быть либо словесное описание проблемы, либо циклограмма (со словесными комментариями). Эти исходные данные разрабатывают соответствующие специалисты в своей области знания. Поскольку алгоритмы становятся все более и более сложными, соответственно растет трудность взаимопонимания программиста и соответствующего специалиста. Согласно [10] в США обратили внимание на то, что программист тратит от 50 до 100% времени на исправление ошибок в программах и их модификацию с целью реализации недостающих вариантов решения задачи.

Если блок-схемы алгоритмов и соответствующие таблицы к ним разработаны специалистом своей области с достаточной аккуратностью у программиста вообще может не возникнуть никаких вопросов, а качество решения задачи будет зависеть только от глубины ее понимания этим специалистом, что вполне закономерно. Разработанные методом структурного программирования “сверху-вниз” алгоритмы обладают в некотором смысле свойством правильности, встроенной в них шаг за шагом. В связи с этим в них, как правило, меньше ошибок и их правильность доказывается легче [10]. Метод в целом прост и доступен для освоения каждым; не требует англоязычной терминологии.

3. Алгоритмы обработки измерительной информации.

Хорошей иллюстрацией метода структурного программирования может быть разработка алгоритмов по вычислению оценок параметров генеральной совокупности значений отдельных измерений (т.е. соответствующего закона распределения). Таких алгоритмов восемь (алгоритм “Выборка” девятый) и решаться они должны поочередно, поскольку в соответствии с теорией погрешностей, результаты вычислений предшествующих алгоритмов используются в последующих. Кроме алгоритма “Размах Выборки” - он не использует результаты предшествующих и может быть задействован независимо ни от какого-либо другого, но его результаты используются во всех последующих. Таким образом, в вычислительном процессе одновременно могут участвовать только два алгоритма. Алгоритм вычисляющий соответствующую оценку использует значения отдельных измерений из алгоритма “Выборка”. Этот процесс имеет циклический характер. В каждом цикле алгоритм “Выборка” выдает очередное значение отдельного измерения, которое используется в алгоритме работающим с ним в паре и снова возврат к алгоритму “Выборка”. По достижении количества циклов равным объему используемой выборки n_0 вычисляется соответствующая оценка. Такой подход позволяет унифицировать алгоритмы с точки зрения использования выборок любых объемов.

Известно, что погрешность вычисляемых оценок зависит от объема используемых выборок и типа закона распределения [3, 5]. Чем больше n_0 тем точнее вычисляемые оценки [1, 3, 5]. В зависимости от предполагаемых объемов используемых выборок n_0 может не иметь смысла реализовать какие-то алгоритмы в качестве программных модулей, кроме алгоритмов “Выборка”, “Среднее” и “Средняя квадратичная погрешность”. Но если допустить, что все девять алгоритмов реализованы, то это не значит, что каждый из них может быть задействован в данный момент. Могут быть, например, повторные включения алгоритмов после коррекции оператором каких-то параметров или в зависимости от объема накопленной выборки. Алгоритмы которые должны быть задействованы задает оператор.

Значения отдельных измерений, принадлежащих одной генеральной совокупности, можно накапливать в базе данных (БД) неодномоментно, если БД и алгоритм “Выборка” рассчитаны на большой объем используемых выборок. Такое накопление не ухудшит соответствующих оценок а только оптимизируют их. При объемах используемых выборок $n_0 \leq 30$ можно задействовать алгоритмы “Выборка”, “Среднее”, “Средняя квадратичная погрешность”, если $n_0 \geq 30$ кроме перечисленных “Контрэксцесс”, “Размах выборки” и “Энтропийный коэффициент”. Чтобы получить более-менее гладкую кривую алгоритм “Закон распределения” можно включать при $n_0 \approx 200 \div 400$ и только после алгоритмов “Размах выборки” и “Энтропийный коэффициент”. Алгоритм “Статистическая устойчивость” тоже требует достаточно больших объемов выборок $n \geq 100$. Исследуя соответствующие параметры реального закона распределения можно оценить зависимость погрешности оценок от объема выборок.

Кроме значений отдельных измерений база данных (БД) также должна содержать массив корректируемых оператором постоянных параметров. Ввод в БД соответствующих массивов с экрана монитора ПЭВМ может быть реализован с использованием стандартных процедур.

3.1. Алгоритм “Выборка” (АВ).

Из базы данных (БД) значения отдельных измерений $x_1, x_2, x_3, \dots, x_{n_0}$ поочередно “извлекаются” настоящим алгоритмом посредством переприсвоения параметру x_i очередного значения, что и отражено в блоках 4 рис. 3.1. Подсчет используемых значений производится счетчиком $n := n + 1$ бл.2, а очередность переприсвоения контролируется блоками опроса счетчика за номером 3. Программа в целом запускается параметром *END*

по ветви “да” бл.1. После выполнения программы параметру END присваивается единица т.е. $END := 1$ и дальше по ветви “нет” бл.1 следует выход из программы.

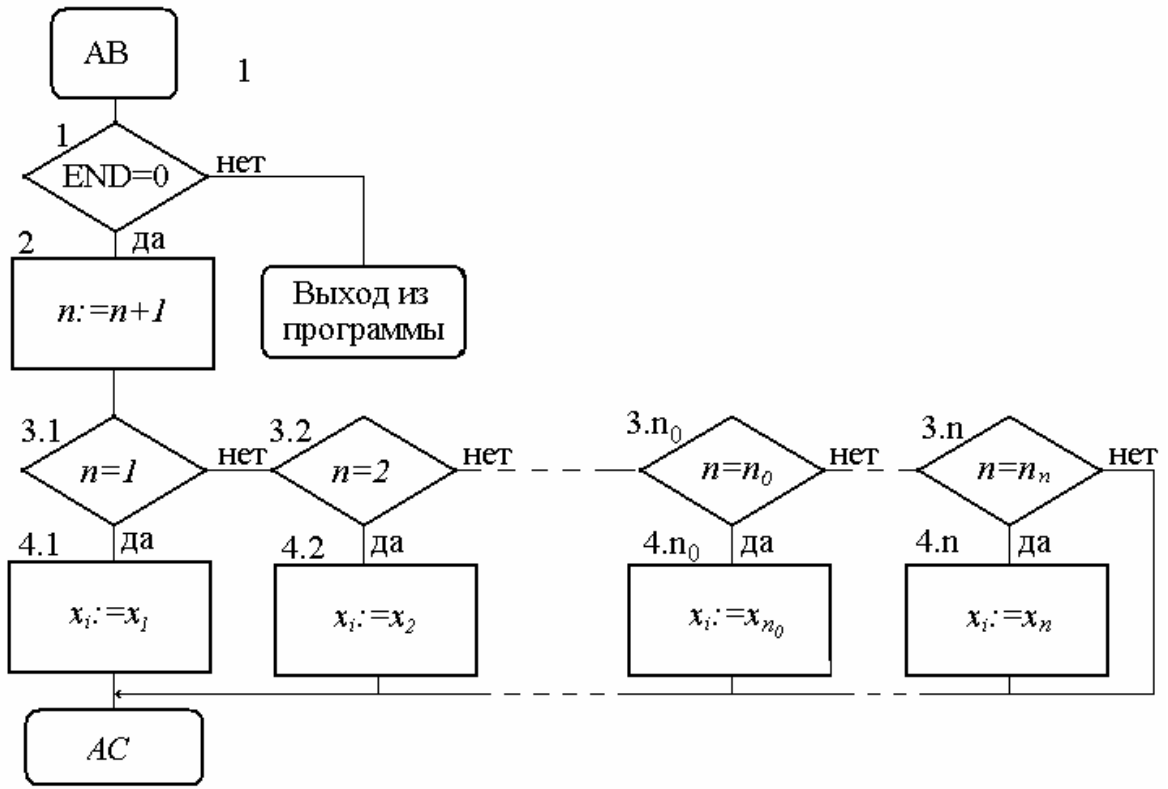


Рис.3.1

После первого переприсвоения параметру $x_i := x_1$ при $n=1$ к настоящему алгоритму подключается алгоритм “Среднее” (AC), который и использует x_i , затем следует возврат к настоящему алгоритму за очередной операцией переприсвоения $x_i := x_2$ с последующим его использованием в алгоритме “Среднее” и т.д. до $n=n_0$, если заданный оператором объем выборки $n_0 < n_n$ т.е. счетчик $n := n + 1$ до конца может и не заполняться.

Таким образом, в один цикл входит операция переприсвоения x_i очередного значения отдельного измерения из БД, затем его использование в подключенном алгоритме с последующим возвратом к очередной операции переприсвоения. После выполнения алгоритма “Среднее” (AC) в нем же обнуляется счетчик $n := 0$ настоящего алгоритма, чем он приводится в исходное состояние для задействования со следующим алгоритмом. Все последующие алгоритмы решаются поочередно, точно также как и алгоритм “Среднее”, используя последовательно формируемые алгоритмом “Выборка” параметры x_i . Общая блок-схема задействования алгоритма “Выборка” и выполнения программы в целом изображена на рис. 3.2, где видно, что каждый алгоритм запускается для выполнения соответствующим признаком из $A1 \div A8$, значения которых $0 \vee 1$ задает оператор их начальной настройкой, кроме того по завершению выполнения каждого алгоритма соответствующему признаку присваивается единица, что обеспечивает переход к выполнению очередного алгоритма или выход из программы.

Таблица переменных параметров АВ

№ п/п	Наименование параметра	Идентификатор	Диапазон изменения	Дискретность	Где формируется	Где используется	Начальная настройка	Примечания
1	Признак выполнения программы	<i>END</i>	0∨1	1	АСУ	АВ	0	
2	Счетчик значений отдельных измерений	<i>n</i>	0÷ <i>n_n</i>	1	АВ, АС, АСКП, АА, АК, АРВ, АЭК, АЗР, АСУ	АВ, АА, АРВ, АЭК, АСУ	0	
3	Значения отдельных измерений	<i>x_{1÷n}</i>			БД	АВ	-	
4	Значения отдельных измерений после переприсвоения	<i>x_i</i>			АВ	АС, АСКП, АА, АК, АРВ, АЭК, АСУ	0	

Таблица постоянных параметров АВ

№ п/п	Наименование параметра	Идентификатор	Значения или диапазон коррекции	Дискретность	Где используется	Примечания
1	Полный объем счетчика отдельных измерений, равный возможному объему выборки в БД.	<i>n_n</i>		1	АВ	
2	Объем используемой Выборки	<i>n₀</i>		1	АА, АРВ, АЭК, АСУ, АВ	Корректируется оператором $n_0 \leq n_n$

Примечание: В таблицах ко всем алгоритмам, там где нет заполненных “Диапазон изменения”, “Дискретности”, “Начальной настройки”, они оцениваются по конкретным экспериментальным данным.

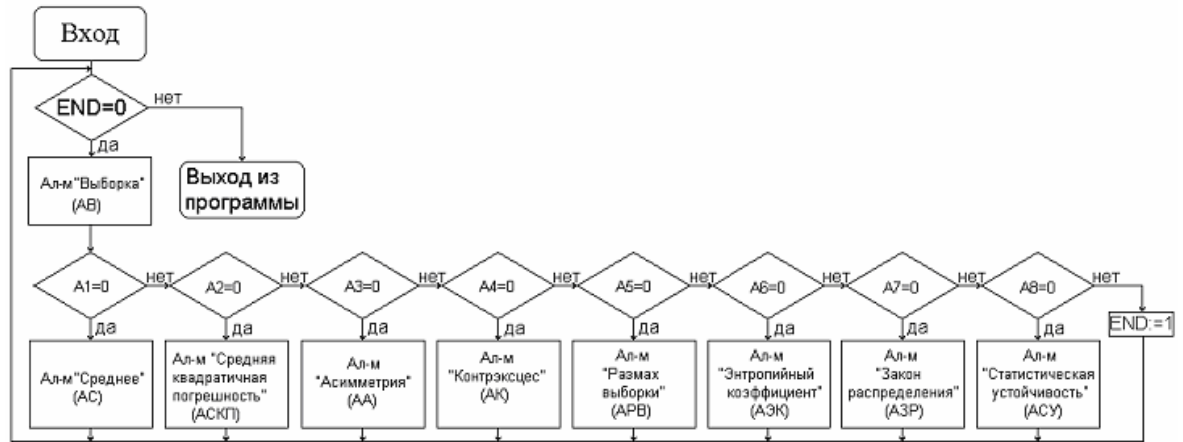


Рис 3.2. Блок-схема задействования алгоритмов, вычисляющих соответствующие оценки с алгоритмом “Выборка” (АВ). Параметры здесь вынесенные за блоки алгоритмов анализируются и формируются в алгоритмах.

3.2. Алгоритм “Среднее” (АС).

Алгоритм вычисляет среднее арифметическое выборки объема n_0 .

$$\bar{x} = \frac{\sum_{i=1}^{n_0} x_i}{n_0}.$$

При больших объемах выборки $\sum_{i=1}^{n_0} x_i$ может принимать неприемлемо большие значения. Чтобы этого не происходило можно предусмотреть автоматическое разбиение выборок большого объема на малые объема M_c , задаваемого оператором, с последующим осреднением средних арифметических малых выборок \bar{x}_M , количество которых B_c должно быть таким чтобы $n_0 = M_c \cdot B_c$.

Алгоритм рис. 3.3 запускается для выполнения по ветви “да” или обходится по ветви “нет” в зависимости от присвоенного значения $0 \vee 1$ параметру $A1$ бл.1.

Формируемые AB в каждом цикле значения параметра x_i суммируются в настоящем алгоритме $\sum x_i := \sum x_i + x_i$ бл.2. и подсчитываются счетчиком $n_M := n_M + 1$ бл.3. До тех пор пока $n_M < M_c$ по ветви “нет” бл.4 происходит возврат к AB за очередным x_i . Если $n_M = M_c$ бл.4 по ветви “да” вычисляются средние арифметические малых выборок $\bar{x}_M := \sum x_i / M_c$ и суммируются $\sum \bar{x}_M := \sum \bar{x}_M + \bar{x}_M$ после чего приводится в исходное сумма $\sum x_i = 0$ и счетчик $n_M := 0$ с целью возобновить суммирование x_i до объема малой выборки M_c . Эти операции отражены в бл.5. Счетчик $n_c := n_c + 1$ бл.6 подсчитывает количество вычисленных средних арифметических малых выборок и если $n_c < B_c$, то по ветви “нет” бл.7 происходит возврат к AB чтобы заново начать суммировать x_i очередной малой выборки. Если количество малых выборок будет равно B_c т.е. $n_c = B_c$ то по ветви “да” бл.7 вычисляется среднее арифметическое выборки объема n_0 $\bar{x} := \sum \bar{x}_M / B_c$ и приводится в исходное сумма $\sum \bar{x}_M := 0$ и счетчик $n_c := 0$. Эти операции отражены в бл.8. Бл.9 завершает выполнение алгоритма обнулением счетчика $n := 0$ алгоритма “Выборка” и формированием признака $A1 := 1$ блокирующего выполнение настоящего алгоритма.

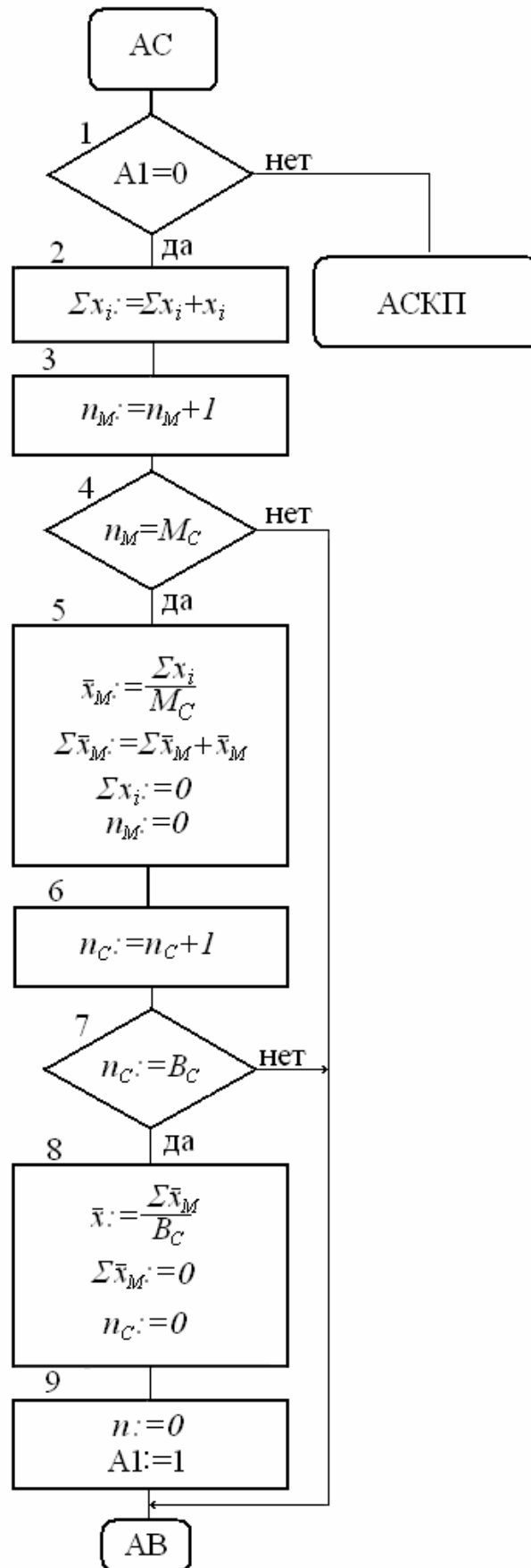


Рис.3.3

Таблица переменных параметров АС

№ п/п	Наименование параметра	Идентификатор	Диапазон изменения	Дискретность	Где формируется	Где используется	Начальная настройка	Примечания
1	Признак блокирующий выполнение алгоритма	A1	0v1	1	АС	АС		Начальную настройку задает оператор
2	Сумма значений отдельных измерений малой выборки	Σx_i			АС	АС	0	
3	Счетчик значений отдельных измерений малой выборки	n_M		1	АС	АС	0	
4	Значения отдельных измерений	x_i			АВ	АС,АС КП,АА, АК,АРВ ,АЭК, АСУ	-	
5	Среднее арифметическое малой выборки	\bar{x}_M			АС	АС	0	
6	Сумма средних арифметических малых выборок	$\Sigma \bar{x}_M$			АС	АС	0	
7	Счетчик малых выборок	n_c		1	АС	АС	0	
8	Среднее арифметическое выборки объема n_0	\bar{x}			АС	АСКП, АА,АК, АЭК, АСУ	0	Документируется
9	Счетчик значений отдельных измерений большой выборки	n		1	АВ	АВ,АС, АСКП, АА, АК, АЭК, АРВ, АСУ	-	

Таблица постоянных параметров АС

№ п/п	Наименование параметра	Идентификатор	Значения или диапазон коррекции	Дискретность	Где используется	Примечания
1	Количество отдельных измерений малой выборки	M_c		1	АС	$M_c \leq n_0$ корректируется оператором
2	Количество малых выборок	B_c		1	АС	$B_c \geq 1$

3.3. Алгоритм “Средняя квадратичная погрешность” (АСКП)

Алгоритм предназначен для вычисления оценки средней квадратичной погрешности по выборке объема n_0 .

$$\sigma(x_i) = \left[\sum_{i=1}^{n_0} (x_i - \bar{x})^2 / n_0 - 1 \right]^{1/2},$$

x_i – значения отдельных измерений,

\bar{x} – среднее арифметическое выборки объема n_0 .

Чтобы избежать накопления неприемлемо больших значений суммы $\sum_{i=1}^{n_0} (x_i - \bar{x})^2$ можно предусмотреть автоматическое разбиение выборок больших объемов на малые объема M_{Π} задаваемого оператором с последующим осреднением средних квадратичных погрешностей малых выборок σ_M , количество которых должно быть таким чтобы $n_0 = M_{\Pi} B_{\Pi}$.

Алгоритм рис. 3.4 выполняется по ветви “да” бл.1 если задаваемая оператором начальная настройка признака $A2 = 0$ и блокируется если $A2 = 1$.

Разности $\Delta x_i := (x_i - \bar{x})$ бл.2 вычисляются в каждом цикле взаимодействия настоящего алгоритма с АВ. Вычисление квадрата этой разности обозначено в бл.3. $R_{\Delta} := (\Delta x_i)^2$, а суммирование R_{Δ} в блоке 4 $\sum R_{\Delta} := \sum R_{\Delta} + R_{\Delta}$. Счетчик $K_M := K_M + 1$ бл.5 подсчитывает количество использованных x_i . Если $K_M < M_{\Pi}$, то по ветви “нет” бл.6 следует переход к алгоритму АВ за очередным x_i , если $K_M = M_{\Pi}$ то по ветви “да” бл.6 вычисляется дисперсия малой выборки $D = \sum R_{\Delta} / (M_{\Pi} - 1)$ и средняя квадратичная погрешность малой выборки $\sigma_M := (D)^{1/2}$ бл.7. σ_M малых выборок суммируется бл.8 $\sum \sigma_M := \sum \sigma_M + \sigma_M$. С целью обработки очередной малой выборки приводятся в исходное параметры $K_M := 0$ и $\sum R_{\Delta} := 0$ бл.9. Счетчиком $n_{\Pi} := n_{\Pi} + 1$ бл.10 подсчитывается количество обработанных малых выборок. Если $n_{\Pi} < B_{\Pi}$ следует возврат к АВ по ветви “нет” бл.11. Если $n_{\Pi} = B_{\Pi}$, то по ветви “да” бл.11 вычисляется средняя квадратичная погрешность выборки объема n_0 $\sigma(x_i) := \sum \sigma_M / B_{\Pi}$ бл.12 и там же обнуляются параметры $n_{\Pi} := 0$ и $\sum \sigma_M := 0$ с целью приведения в исходное алгоритма в целом. В бл.13 отражены обнуление счетчика отдельных измерений в АВ $n := 0$ и

присвоение $A_2:=1$, чем блокируется выполнение настоящего алгоритма и переход к выполнению очередного или выход из программы.

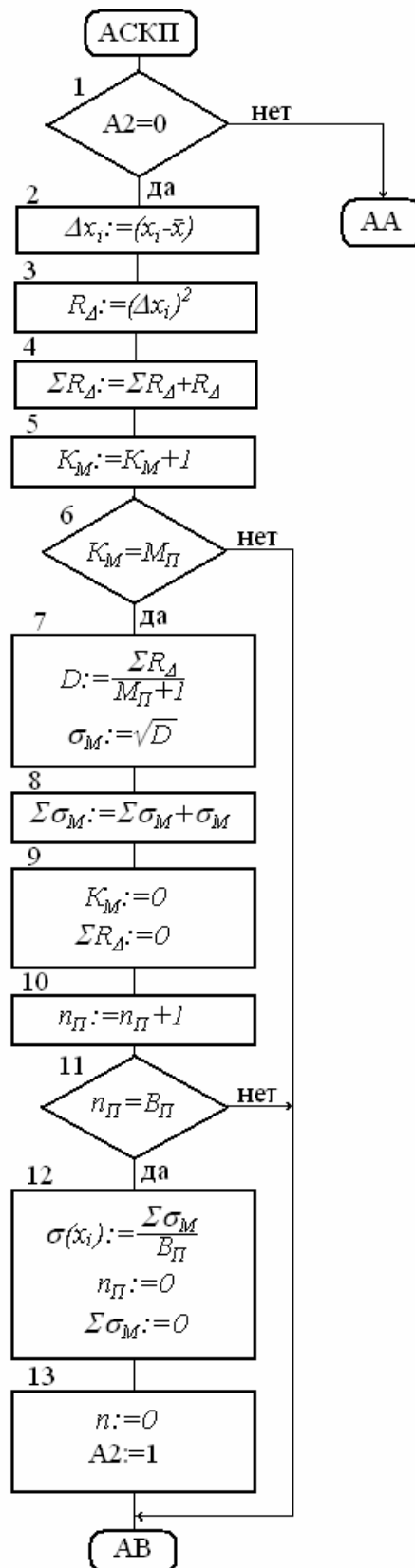


Рис.3.4.

Таблица переменных параметров АСКП.

№ п./п	Наименование Параметра	Идентификатор	Диапазон изменения	Дискретность	Где формируется	Где используется	Начальная настройка	Примечания
1	Признак блокирования алгоритма	A2	0÷1	1	АСКП	АСКП		Нач. настройка задается оператором.
2	Значения отдельных измерений	x_i			AB	AC, АСКП, AA,AK, APB, AЭC, ACY	–	
3	Среднее арифметическое выборки объема n_0	\bar{x}			AC	АСКП, AA, AK, ACY	–	
4	Разность значений отдельных измерений и значения среднего арифметического	Δx_i			АСКП	АСКП	0	
5	Квадрат разности Δx_i	R_{Δ}			АСКП	АСКП	0	
6	Сумма квадратов разностей R_{Δ}	$\sum R_{\Delta}$			АСКП	АСКП	0	
7	Счетчик x_i малой выборки	K_M		1	АСКП	АСКП	0	
8	Дисперсия малой выборки	D			АСКП	АСКП	0	
9	Средняя квадратичная погрешность малой выборки	σ_M			АСКП	АСКП	0	
10	Сумма σ_M	$\sum \sigma_M$			АСКП	АСКП	0	
11	Счетчик количества вычисленных σ_M	n_{Π}		1	АСКП	АСКП	0	
12	Средняя квадратичная погрешность выборки объема n_0	$\sigma(x_i)$			АСКП	АСКП	0	Документируется
13	Счетчик отдельных измерений алгоритма “Выборка”	n		1	AB,AC, АСКП, AA,AK, APB, ACY	AA,APB, AЭK, ACY	0	

Таблица постоянных параметров АСКП

№ п/п	Наименование параметра	Идентификатор	Значения или диапазон коррекции	Дискретность	Где используется	Примечания
1	Объем малой выборки	M_{II}		1	АСКП	Задается оператором $M_{II} \leq n_0$
2	Количество малых выборок	B_{II}		1	АСКП	$B_{II} \geq 1$

3.4. Алгоритм “Асимметрия” (АА)

Алгоритм рис. 3.5 предназначен для вычисления оценки асимметрии закона распределения по выборке объема n_0 ,

$$a(x_i) = \mu_a(x_i) / \sigma^3(x_i),$$

где

$$\mu_a(x_i) = 1/n_0 \cdot \sum_{i=1}^{n_0} (x_i - \bar{x})^3.$$

$\sigma(x_i)$ - выборочная средняя квадратичная погрешность.

x_i - значение отдельного измерения.

\bar{x} - среднее арифметическое выборки объема n_0 .

Для симметричных законов распределения $a(x_i)$ близка к нулю. Разности $(x_i - \bar{x})$ суммируются с учетом их знака и суммы $\sum_{i=1}^{n_0} (x_i - \bar{x})^3$ не могут иметь больших значений.

Признак $A3$ запускает выполнение алгоритма по ветви ‘да’ бл.1, если его начальная настройка $A3 = 0$ и блокирует если $A3 = 1$. Его начальную настройку задает оператор.

В бл.2 отражено вычисление в каждом цикле разности $\Delta x_a := x_i - \bar{x}$, а в бл.3 возведение этой разности в куб т.е. $R_a := (\Delta x_a)^3$ и суммирование $\sum R_a := \sum R_a + R_a$ в каждом цикле. Контроль за количеством использованных x_i отражается в бл.4, посредством опроса счетчика отдельных измерений в алгоритме “Выборка”. Если $n < n_0$, то следует возврат к AB по ветви “нет” бл.4 за очередным x_i . Если $n = n_0$, то вычисляются параметры $\mu_a(x_i) := \sum R_a / n_0$, $C_a := \sigma^3(x_i)$ и асимметрия $\alpha(x_i) := \mu_a(x_i) / C_a$ бл.5. Обнулением $\sum R_a := 0$ алгоритм приводится в исходное состояние бл.6. В бл.7 обнуляется счетчик AB $n := 0$ и формируется признак $A3 := 1$ блокирующий выполнение алгоритма.

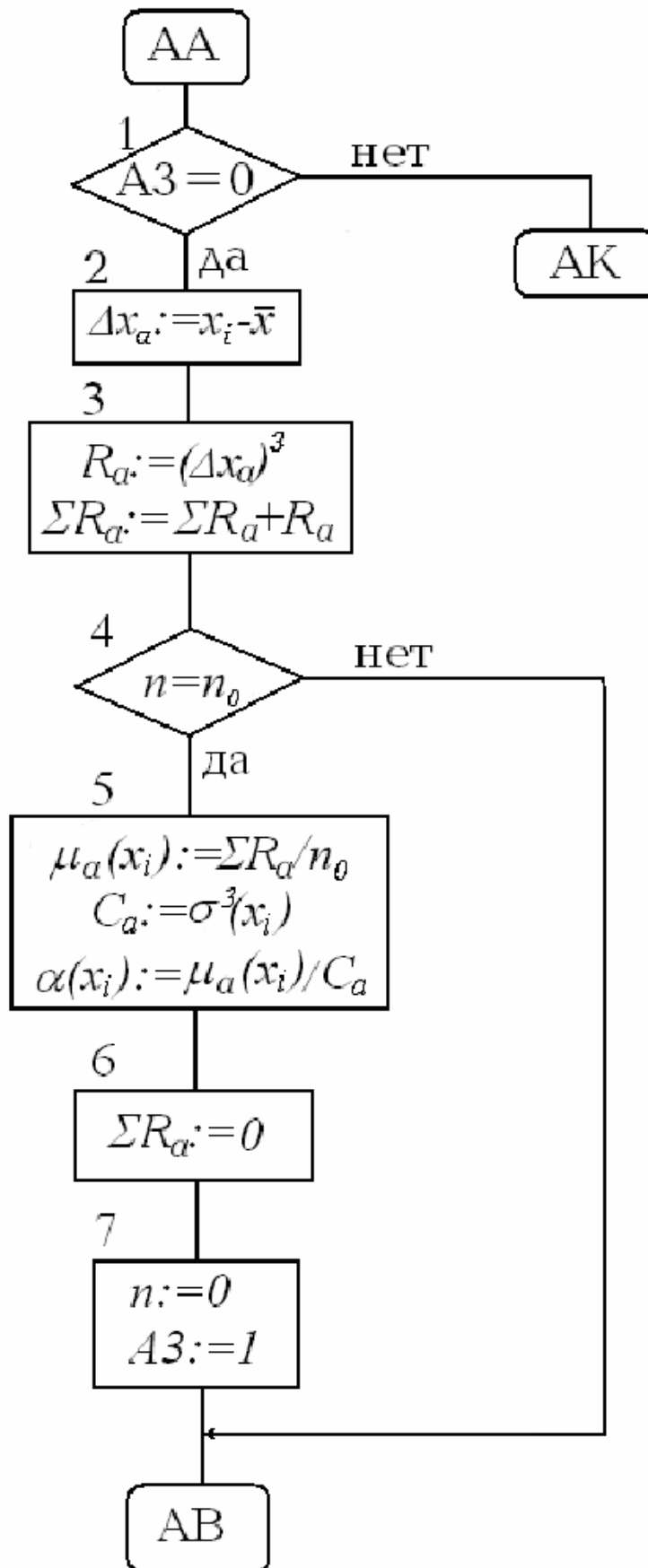


Рис.3.5.

Таблица переменных параметров АА

№ п/п	Наименование Параметра	Идентификатор	Диапазон изменения	Дискретность	Где формируется	Где используется	Начальная настройка	Примечания
1	Признак блокирования алгоритма	A3	0~1	1	АА	АА		Нач.настройку задает оператор
2	Значения отдельных Измерений	x_i			АВ	АС,АСКП, АА,АК, АРВ,АЭК, АСУ	–	
3	Среднее арифметическое выборки объема n_0	\bar{x}			АС	АСКП,АА, АК,АЭК, АСУ	–	
4	Разность значений отдельных измерений и среднего арифметического	Δx_a			АА	АА	0	
5	Счетчик отдельных измерений алгоритма “Выборка”	n		1	АВ	АВ,АС,АС КП,АА,АК, АРВ,АЭК, АСП,АСУ	0	
6	Оценка третьего центрального момента закона распределения	$\mu_a(x_i)$			АА	АА	0	
7	Средняя квадратичная погрешность выборки объема n_0	$\sigma(x_i)$			АСКП	АА,АК, АЭК,АСУ	–	
8	Куб СКП	$C\alpha$			АА	АА	0	
9	Асимметрия	$\alpha(x_i)$			АА		0	Документируется

Таблица постоянных параметров АА

№ п/п	Наименование параметра	Идентификатор	Значения или диапазон коррекции	Дискретность	Где используется	Примечания
1	Объем используемой Выборки	n_0		1	АРВ,АЭК, АСУ	Задается оператором

3.5. Алгоритм “Контрэкссесс” (АК).

В процессе вычисления оценки контрэкссесса $\mathfrak{x}(x_i)$ необходимо вычислять сумму $\sum_{i=1}^{n_0} (x_i - \bar{x})^4$, которая может возрастать очень быстро в зависимости от объема выборки и типа закона распределения и достигать неприемлемых значений, поэтому может оказаться, что рациональнее вычислять контрэкссесс с использованием малых выборок объема \bar{n} с последующим суммированием \mathfrak{x}_M и осреднением по количеству обработанных малых выборок $n_{кэ}$. Последовательность вычисления оценок контрэкссесса по малым выборкам может быть (см. разд. 1.2) следующей:

$$\mu_M = \frac{1}{\bar{n}} \sum_{i=1}^{\bar{n}} (x_i - \bar{x})^4,$$

$$\text{поправки } C_1 = \frac{\bar{n}(\bar{n}^2 - 2\bar{n} + 3)}{(\bar{n} - 1)(\bar{n} - 2)(\bar{n} - 3)}, \quad C_2 = \frac{3(\bar{n} - 1)(2\bar{n} - 3)}{\bar{n}(\bar{n} - 2)(\bar{n} - 3)}, \quad C_3 = C_2 \sigma^2(x_i), \quad \bar{\mu}_M = C_1 \mu_M - C_3,$$

$$\mathfrak{x}_M = \sigma^2(x_i) / \sqrt{\bar{\mu}_M}.$$

Если объем большой выборки n_0 , а число обрабатываемых малых выборок $n_{кэ}$, то должно быть $n_0 = \bar{n} \cdot n_{кэ}$. Параметры \bar{n} и $n_{кэ}$ выбирает и корректирует оператор с учетом конкретных условий.

Оператор может блокировать выполнения алгоритма рис.3.6. начальной настройкой параметра $A4=1$ по ветви “нет” бл.1. Если $A4=0$ алгоритм выполняется по ветви “да”.

В бл.2 отражено вычисление разности значений отдельных измерений x_i и среднего арифметического \bar{x} в каждом цикле взаимодействия с алгоритмом “Выборка” т.е. $\Delta x_k = x_i - \bar{x}$. Эти разности возводятся в четвертую степень $R_k := (\Delta x_k)^4$ и суммируются $\sum R_k := \sum R_k + R_k$ в каждом цикле бл.3. Счетчик $S_k := S_k + 1$ бл.4 подсчитывает количество использованных x_i . Если $S_k < \bar{n}$ то по ветви “нет” бл.5 происходит возврат к алгоритму “Выборка” (АВ) за очередным x_i . Если $S_k = \bar{n}$ то по ветви “да” бл.5 производится анализ состояния признака F, бл.6. Начальная настройка этого признака всегда ноль, поэтому по ветви “да” бл.6 разрешается разовое вычисление, постоянных на время выполнения алгоритма, параметров C_1, C_2, C_3 бл.7, после чего параметру F присваивается единица т.е. $F:=1$ бл.7 чем блокируется повторное вычисление этих параметров по ветви “нет” бл.6 при дальнейшем выполнении алгоритма.

Если $S_k = \bar{n}$ бл.5 то это означает, что разрешены операции вычисления параметров малой выборки $\mu_M, \bar{\mu}_M, \mathfrak{x}_M$ и суммирование $\sum \mathfrak{x}_M := \sum \mathfrak{x}_M + \mathfrak{x}_M$ после обработки очередной малой выборки – эти операции отражены в бл.8. Бл.9 отражает приведение в исходное параметров $\sum R_k := 0$ и $S_k := 0$ с целью обработки очередной малой выборки. Счетчик $n_k := n_k + 1$ бл.10 подсчитывает количество обработанных малых выборок и если $n_k < n_{кэ}$ то по ветви “нет” бл.11 происходит возврат к алгоритму “Выборка” (АВ) и начинается обработка очередной малой выборки.

Если $n_k = n_{кэ}$ то по ветви “да” бл.11 вычисляется контрэкссесс выборки объема n_0 $\mathfrak{x}(x_i) := \sum \mathfrak{x}_M / n_{кэ}$ бл.12 и дальше приводятся в исходное параметры алгоритма $\sum \mathfrak{x}_M := 0$ $n_k := 0$, $F := 0$.

В бл.13 отражены обнуление счетчика отдельных измерений в алгоритме “Выборка” $n := 0$ и формируется параметр $A4:=1$ с целью отключения алгоритма.

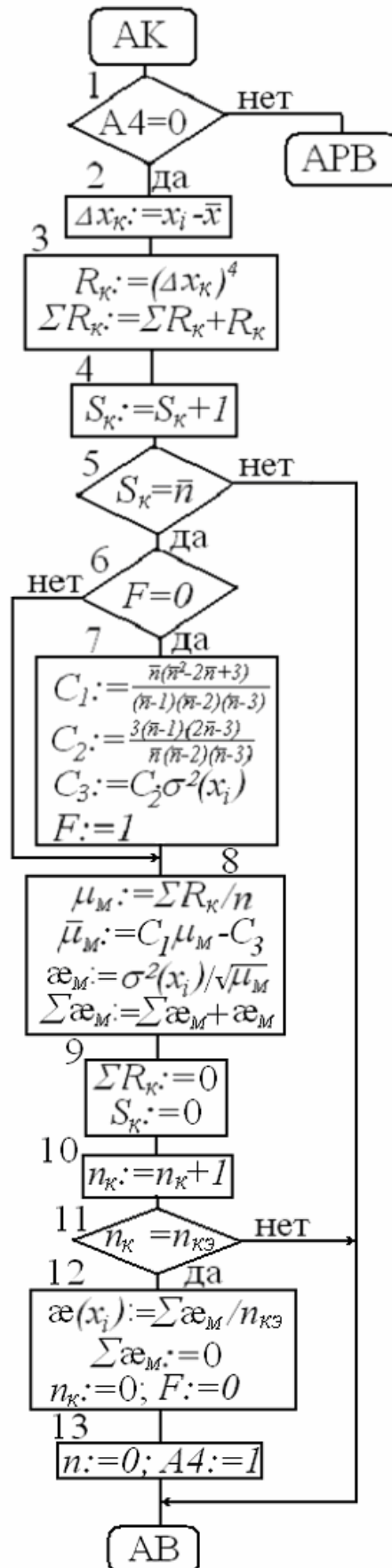


Рис.3.6.

Таблица переменных параметров АК.

№ п/п	Наименование параметра	Идентификатор	Диапазон изменения	Дискретность	Где формируется	Где используется	Начальная настройка	Примечания
1	Признак блокирования выполнения алгоритма	A4	0∨1	1	АК	АК		Начальная настройка задается оператором
2	Значения отдельных измерений	x_i			AB	AC,ACKП, AA,AK, APB,AEK, ACY	-	
3	Среднее арифметическое выборки объема n_0	\bar{x}			AC	ACKП,AA, AK,ACY		
4	Разность значений отдельных измерений и среднего арифметического	Δx_k			АК	АК	0	
5	Δx_k в четвертой степени	R_k			АК	АК	0	
6	Сумма R_k	$\sum R_k$			АК	АК	0	
7	Счетчик отдельных измерений малой выборки	S_k			АК	АК	0	
8	Признак блокирования разового вычисления параметров C_1, C_2, C_3 после их вычисления	F	0∨1	1	АК	АК	0	
9	Оценка четвертого центрального момента без поправок	μ_M			АК	АК	0	
10	Оценка четвертого центрального момента с поправками	$\bar{\mu}_M$			АК	АК	0	

1	2	3	4	5	6	7	8	9
11	Оценка контрэкссесса по малой выборке	\bar{x}_M			АК	АК	0	
12	Сумма значений \bar{x}_M малых выборок	$\sum \bar{x}_M$			АК	АК	0	
13	Счетчик количества малых выборок	n_k		1	АК	АК	0	
14	Оценка контрэкссесса по выборке объема n_0	$\bar{x}(x_i)$			АК		0	
15	Оценка средней квадратичной погрешности по выборке объема n_0	$\sigma(x_i)$			АС КП	АА, АК АСУ	–	
16	Счетчик используемых значений отдельных измерений	n			АВ	АС, АСКП, АРВ, АЭК АЗР, АСУ, АК, АА	0	
17	Поправочный коэффициент-1	C_1			АК	АК	0	
18	Поправочный коэффициент-2	C_2			АК	АК	0	
18	Поправочный коэффициент-3	C_3			АК	АК	0	

Таблица постоянных параметров АК

№ п/п	Наименование параметра	Идентификатор	Значения или диапазон коррекции	Дискретность	Где используется	Примечания
1	Объем малой выборки	\bar{n}		1	АК	$\bar{n} \leq n_0$ корректируется оператором
2	Количество используемых малых выборок	$n_{кэ}$		1	АК	$n_{кэ} \geq 1$

3.6. Алгоритм “Размах выборки” (АРВ).

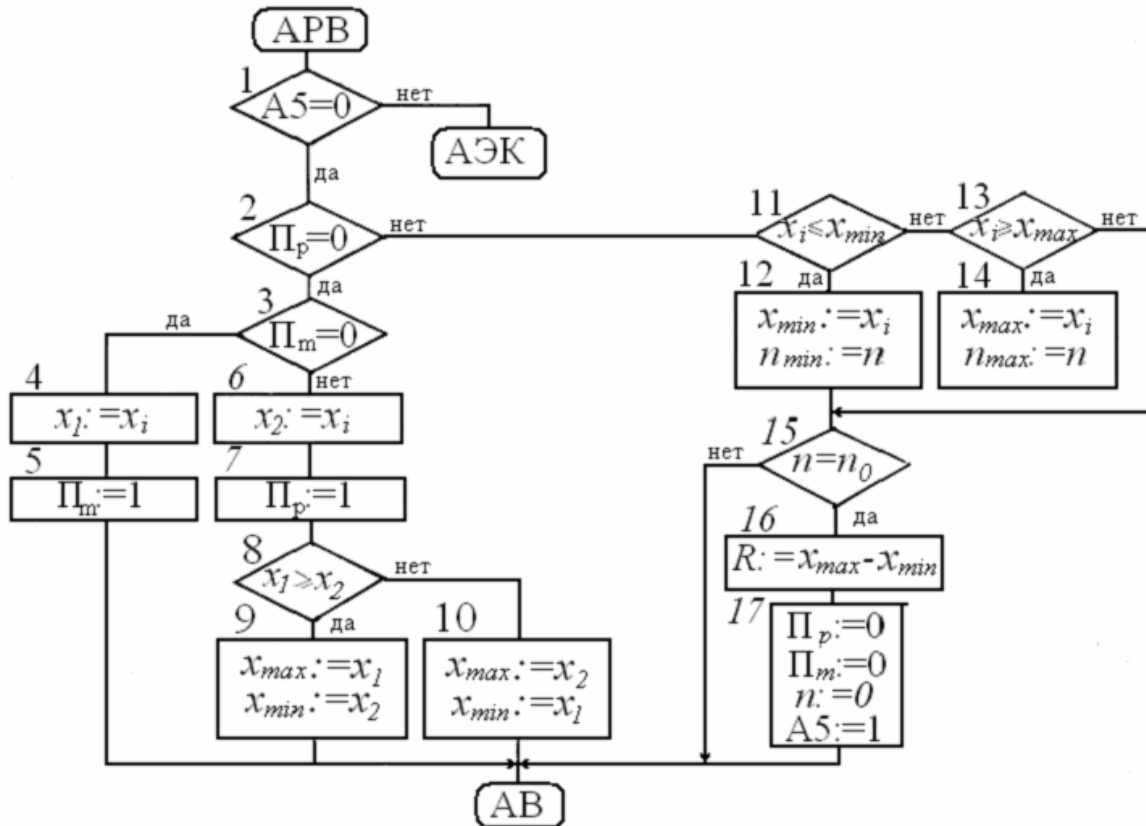


Рис.3.7.

Алгоритм вычисляет размах выборки объема n_0 $R = x_{\max} - x_{\min}$, для чего нужно найти в выборке наибольшее x_{\max} и наименьшее x_{\min} значения отдельных измерений.

Признак $A5$ в алгоритме рис. 3.7 блокирует выполнение алгоритма, если в начальной настройке $A5=1$, по ветви “нет” бл.1. Алгоритм выполняется если $A5=0$ по ветви “да” бл.1.

Признак P_p бл.2 делит алгоритм на две части выполняющие разные задачи. Первая часть по ветви “да” бл.2 ($P_p=0$) должна запомнить два первых в выборке значения отдельных измерений x_1 и x_2 , сравнить их и большему присвоить x_{\max} а меньшему x_{\min} . Запоминание разделено признаком P_m бл.3. По ветви “да” бл.3 ($P_m=0$) $x_1 := x_i$ бл.4 и дальше $P_m=1$ бл.5 чем блокируется заход на эту ветвь и дальше следует выход из алгоритма к АВ за вторым x_i . При втором заходе в алгоритм поскольку $P_m=1$ то по ветви “нет” бл.3 выполняется операция присвоения $x_2 := x_i$ бл.6 и признаку $P_p:=1$ бл.7 чем будет блокироваться эта часть алгоритма при следующих заходах в него. Операция сравнения x_1 и x_2 отражена в бл.8. Если $x_1 \geq x_2$, то по ветви “да” $x_{\max} := x_1$, $x_{\min} := x_2$ бл.9, а если $x_1 < x_2$ то по ветви “нет” $x_{\max} := x_2$ а $x_{\min} := x_1$ бл.10, после чего следует выход к алгоритму АВ за третьим x_i .

Вторая часть алгоритма выполняется по ветви “нет” бл.2 (поскольку $P_p=1$), где сравниваются все последующие в выборке x_i с x_{\min} и x_{\max} . Если $x_i \leq x_{\min}$ то по ветви “да” бл.11 происходит операция переприсвоения x_{\min} очередного x_i т.е. $x_{\min} := x_i$ бл.12 и запоминается его порядковый номер в выборке $n_{\min} := n$. Если очередное $x_i > x_{\min}$ то по

ветви “нет” x_i сравнивается с x_{\max} бл.13. При $x_i \geq x_{\max}$ производится операция переприсвоения $x_{\max} := x_i$ бл.14 и запоминается порядковый номер x_{\max} т.е. $n_{\max} := n$. Случай $x_i < x_{\max}$ бл.13 будет означать, что очередное x_i попало в интервал $x_{\min} < x_i < x_{\max}$, поэтому операции переприсвоения не происходит по ветви “нет” бл.13. Завершение выше перечисленных операций в одном цикле приводит к опросу счетчика $n := n + 1$ в алгоритме “Выборка” – это бл.15. Если $n < n_0$ где n_0 объем используемой выборки, то по ветви “нет”, бл.15 следует выход из алгоритма к АВ за очередным x_i . Если $n = n_0$ то по ветви “да”, бл.15 следует вычисление размаха выборки $R = x_{\max} - x_{\min}$ бл.16 и дальше приведение в исходное параметров алгоритма $П_p := 0$, $П_m := 0$, обнуление счетчика отдельных измерений в АВ $n := 0$ и присвоение А5:=1 чем блокируется заход в настоящий алгоритм и переход к выполнению очередного или выход из программы если все последующие алгоритмы заблокированы для выполнения.

Таблица переменных параметров АРВ

№ п/п	Наименование параметра	Идентификатор	Диапазон изменения	Дискретность	Где формируется	Где используется	Начальная настройка	Примечания
1	2	3	4	5	6	7	8	9
1	Признак блокирования выполнения алгоритма	А5	0∨1	1	АРВ	АРВ		Начальную настройку выбирает оператор
2	Признак блокирующий ту часть алгоритма, которая обрабатывает первые в выборке два x_i	П _p	0∨1	1	АРВ	АРВ	0	
3	Признак обеспечивающий Запоминание значений первых двух x_i	П _m	0∨1	1	АРВ	АРВ	0	
4	Значение первого в выборке отдельного измерения	x_1			АРВ	АРВ	0	
5	Значение второго в выборке отдельного измерения	x_2			АРВ	АРВ	0	

1	2	3	4	5	6	7	8	9
6	Значения отдельных измерений формируемых в АВ	x_i			АВ	АС, АСКП, АААК, АРВ, АЭК АСУ	–	
7	Наибольшее в выборке значение отдельного измерения	x_{\max}			АРВ	АРВ	0	Документируется
8	Наименьшее в выборке значение отдельного измерения	x_{\min}			АРВ	АРВ	0	Документируется
9	Порядковый номер x_{\max} в выборке	n_{\max}			АРВ	АРВ	0	Документируется
10	Порядковый номер x_{\min} в выборке	n_{\min}			АРВ	АРВ	0	Документируется
11	Счетчик отдельных измерений в алгоритме “Выборка”	n		1	АВ	АС, АСКП, АА, АК, АРВ, АЭК АЭР, АСУ	0	
12	Размах выборки	R			АРВ	АЭК, АСУ	0	Документируется

Таблица постоянных параметров АРВ

№ п/п	Наименование параметра	Идентификатор	Значение или диапазон коррекции	Дискретность	Где используется	Примечания
1	Объем используемой выборки	n_0		1	АА, АРВ АЭК, АСУ	Документируется

3.7. Алгоритм “Энтропийный коэффициент” (АЭК).

Алгоритм вычисляет оценку энтропийного коэффициента закона распределения по выборке объема n_0 , что поможет идентифицировать его тип [3, 6, 7]

$$K_3 = \Delta_3(x_i) / \sigma(x_i),$$

$\Delta_3(x_i)$ - оценка энтропийного значения погрешности,

$\sigma(x_i)$ - оценка средней квадратичной погрешности.

Оценка энтропийного значения погрешности вычисляется по выборке объема n_0 как

$$\Delta_3(x_i) = \frac{dn_0}{2} 10^{-\frac{1}{n_0} \sum_{i=1}^{n_0} n_i \lg n_i}.$$

d – ширина интервала, на которые разбивается размах выборки R (см. рис. 3.9),
 n_i – количество значений отдельных измерений попадающих в i -ый интервал,
 m_0 – количество интервалов d укладываемых в размах выборки R .

Если обозначить $\alpha = dn_0 / 2$ а $\sum Z = \sum_{i=1}^{n_0} n_i \lg n_i$, то

$$\Delta_3(x_i) = \alpha \cdot 10^{-\frac{1}{n_0} \sum Z},$$

и дальше если $\beta = \frac{1}{n_0} \sum Z$, то

$$\Delta_3(x_i) = \alpha \cdot 10^{-\beta}.$$

С учетом поправки на недостаточно большое число n_i попадающих в каждый интервал d [3]

$$\Delta_3^0(x_i) = \Pi_1 \Delta_3(x_i),$$

$$\Pi_1 = \left(1 + \frac{m_0}{2n_0} \right).$$

Для нормального закона распределения $\Delta_3^0 = \frac{1}{2} e^{4,132\sigma}$ [3,7].

Признаком А6 по ветви “нет” бл.1 (Рис.3.8) блокируется выполнение алгоритма с последующим переходом на выполнение следующего алгоритма АЗР, если оператором в начальной настройке присваивается А6:=1 Если А6=0 алгоритм выполняется по ветви “да” бл.1

На первом заходе в алгоритм начальная настройка признана $\Pi_3 = 0$, поэтому по ветви “да” бл.2 выполняются разовые вычисления постоянных параметров алгоритма d , α , Π_1 бл.3, которые зависят от используемого на данный момент объема выборки n_0 и корректируемого оператором количества интервалов m_0 шириной d , которые оцениваются исходя из конкретных экспериментальных данных с учетом чувствительности используемого средства измерений.

После выполнения разовых вычислений признаку $\Pi_3 := 1$ бл.4, чем блокируется повторное вычисление выше обозначенных параметров по ветви “нет” бл.2 при последующих заходах в алгоритм.

Чтобы вычислить энтропийное значение погрешности, и соответственно энтропийный коэффициент закона распределения, нужно исходя из формулы для Δ_3 найти n_i т.е. количество значений отдельных измерений x_i попадающих в каждый интервал шириной d .

Подсчет n_i , количества значений отдельных измерений x_i попавших в каждый из интервалов $1, 2, 3, \dots, m_0$ рис. 3.9 шириной d , начинается с вычисления координат соответствующих интервалов (по оси x) $x_j := x_{\min} + md$ и $x_{j+1} := x_{\min} + (m+1)d$ бл.5. Начальная настройка счетчика $m := m+1$ бл.6 $m=0$ поэтому координаты первого интервала $x_j := x_{\min}$ и $x_{j+1} := x_{\min} + d$, если x_i после проверки бл.7 не попадает в первый интервал, то по ветви “нет” бл.7 через бл.12 по ветви “да” при $m \leq m_0$ происходит возврат к бл.5 и вычисление координат следующего интервала с учетом того, что на счетчике $m := m+1$ бл.6 уже не ноль а единица т.е. $m=1$, поэтому координаты следующего интервала будут $x_j := x_{\min} + d$ и $x_{j+1} := x_{\min} + 2d$ и дальше прибавление единицы счетчика $m := m+1$ (т.е. $m=2$) и снова проверка на попадание x_i во второй интервал.

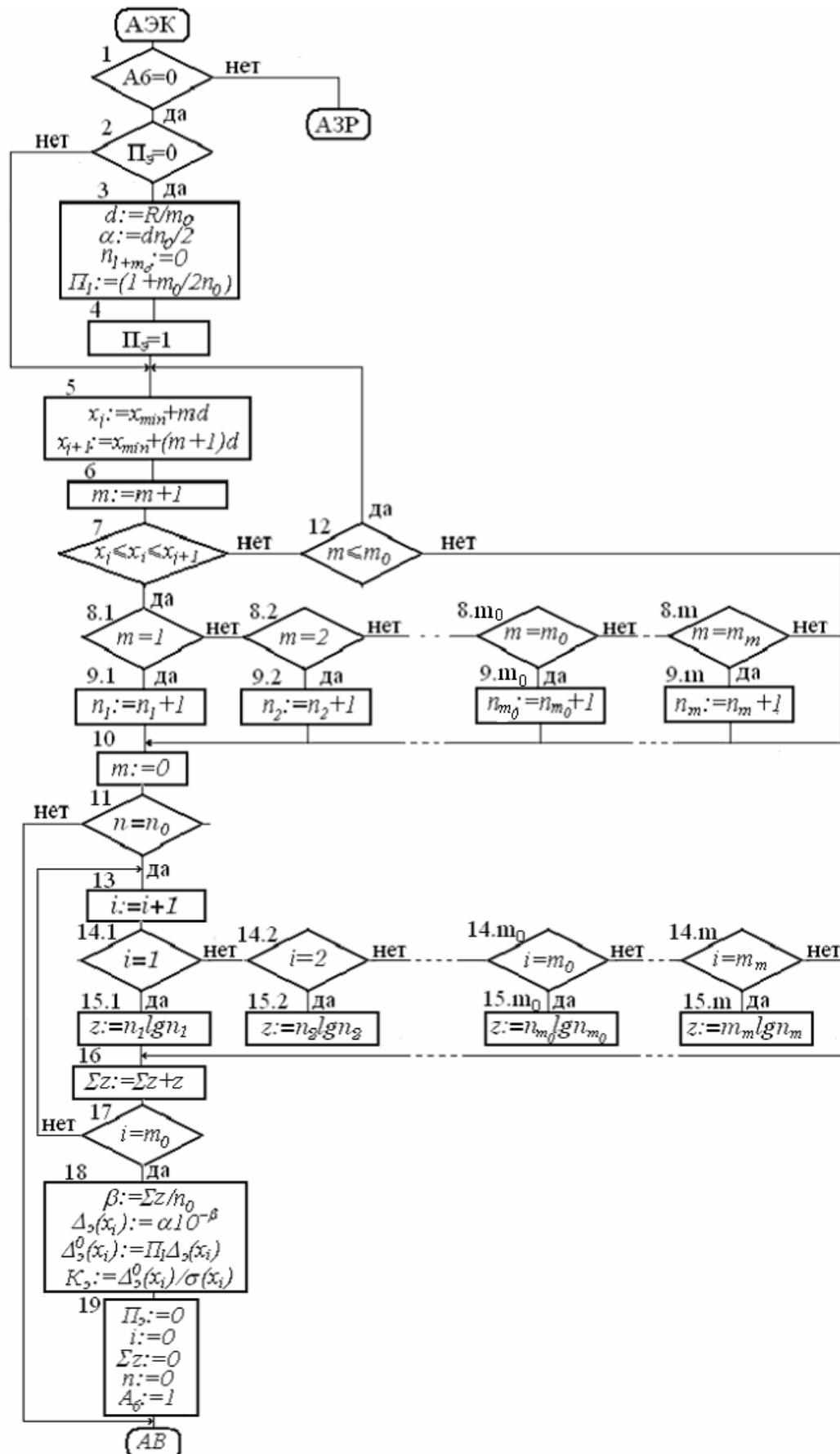


Рис.3.8.

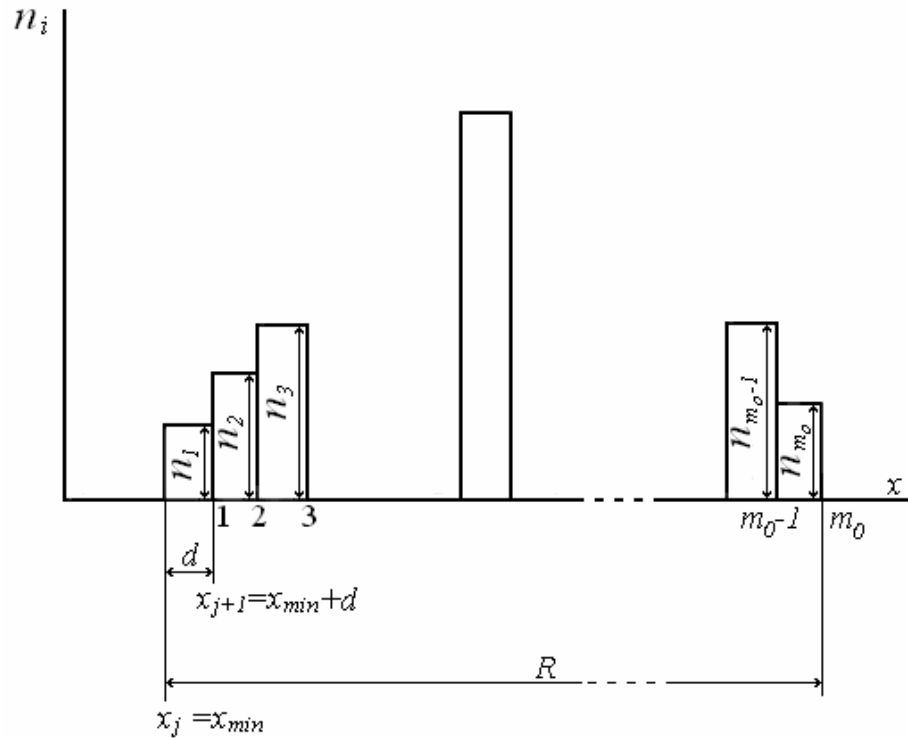


Рис. 3.9

Таким образом, при последовательной проверке, до $m=m_0$ по ветви “нет” бл.7 и по ветви “да” бл.12, попадания x_i в интервалы $x_j \div x_{j+1}$ обнаружится интервал куда x_i обязательно попадет и тогда по ветви “да” бл.7 после опроса номера этого интервала блоки 8 прибавляется единица соответствующего счетчика блоки 9, а дальше обнуляется счетчик интервалов $m:=0$ бл.10 и если количество использованных x_i не достигло n_0 то по ветви “нет” бл.11 следует возврат к АВ за очередным x_i , чтобы начать проверку на попадание с первого интервала.

По завершении подсчета n_i в интервалах $1, 2, \dots, m_0$, т.е. когда количество проверяемых x_i стало равным $n = n_0$ то по ветви “да” бл.11 необходимо вычислить $\sum Z := \sum_{i=1}^{m_0} n_i \lg n_i$.

Счетчик $i:=i+1$ бл.13 подсчитывает суммируемые $n_i \lg n_i$. Блоки 14 опрашивают счетчик i бл.13 на предмет какое из очередных вычислений $n_i \lg n_i$ ($i=1 \div m_0$) необходимо произвести, что и реализуется в одном из соответствующих блоков 15, после чего каждое последовательно вычисленное $n_i \lg n_i$ суммируется бл.16 $\sum Z := \sum Z + Z$. Если количество суммируемых Z не достигло $i = m_0$ бл.17 т.е. $i < m_0$, то по ветви “нет” бл.17 к бл.13 с последующим вычислением очередного значения $Z := n_i \lg n_i$. Как только количество вычисляемых и суммируемых Z достигнет m_0 т.е. $i = m_0$ бл.17 то по ветви “да” будут произведены вычисления обозначенные в бл.18 в том числе и энтропийный коэффициент $K_s := \Delta^0(x_i) / \sigma(x_i)$.

В заключение алгоритм приводится в исходное состояние, посредством обнуления соответствующих параметров бл.19 т.е. $\Pi_s := 0$, $\sum Z := 0$, $i := 0$, кроме счетчиков $n_{1 \div m_0}$, результаты подсчета которых используются в АЗР. Обнуляется также счетчик x_i в АВ $n := 0$ и блокируется выполнение настоящего алгоритма присвоением А6:=1.

Таблица переменных параметров АЭК

№ п/п	Наименование параметра	Идентификатор	Диапазон изменения	Дискретность	Где формируется	Где используется	Начальная настройка	Примечания
1	2	3	4	5	6	7	8	9
1	Признак блокирующий выполнение алгоритма	А6	0∨1	1	АЭК	АЭК		Нач.настройка задается оператором
2	Признак блокирующий разовые вычисления	П ₃	0∨1	1	АЭК	АЭК	0	
3	Размах выборки объема n_0	R			АРВ	АЭК, АСУ	–	
4	Ширина интервала на который разбит размах выборки	d			АЭК	АЭК, АЗР	0	Документируется
5	Переобозначене-1 в формуле для Δ_j	α			АЭК	АЭК	0	
6	Поправка для Δ_j^l	П ₁			АЭК	АЭК	0	
7	Начальная координата интервалов шириной d	x_j			АЭК	АЭК	0	
8	Конечная координата интервалов шириной d	x_{j+1}			АЭК	АЭК	0	
9	Счетчик интервалов шириной d	m		1	АЭК	АЭК	0	
10	Значения отдельных измерений	x_i			АВ	АС, АСКП, АА, АК, АРВ, АЭК, АСУ	–	
11	Счетчики количества значений отдельных измерений попадающих в интервалы шириной d	$n_{1 \div m_m}$		1	АЭК	АЭК, АЗР	0	Документируется
12	Счетчик обеспечивающий последовательное вычисление и суммирование $\sum n_i \lg n_i$	i		1	АЭК	АЭК	0	
13	Последовательно вычисляемое для каждого интервала $n_i \lg n_i$	Z			АЭК	АЭК	0	

1	2	3	4	5	6	7	8	9
14	Сумма параметров Z	$\sum Z$			АЭК	АЭК	0	
15	Переобозначение-2 в формуле для Δ_3	β			АЭК	АЭК	0	
16	Оценка энтропийного значения погрешности без поправки Π_1	$\Delta_3(x_i)$			АЭК	АЭК	0	Документируется
17	Оценка энтропийного значения погрешности с учетом поправки	$\Delta_3'(x_i)$			АЭК	АЭК	0	Документируется
18	Оценка энтропийного коэффициента по выборке n_0	K_3		10^{-3}	АЭК		0	Документируется
19	Счетчик значений отдельных измерений в алгоритме "Выборка"	n		1	АВ	АС, АСКП, АА,АК, АРВ, АЭК АСУ, АЗР		

Таблица постоянных параметров АЭК

№ п/п	Наименование параметра	Идентификатор	Значение или диапазон коррекции	Дискретность	Где используется	Примечания
1	Реализуемое количество интервалов в программе	m_m		1	АЭК	Оценивается с запасом
2	Количество используемых интервалов шириной d	m_0		1	АЭК,АЗР	$m_0 \leq m_m$
3	Объем используемой выборки	n_0		1	АА,АРВ, АЭК,АСУ	Корректируется оператором

3.8. Алгоритм "Закон распределения" (АЗР)

Алгоритм рис. 3.10 вычисляет оценки координат точек кривой закона распределения. Плавную кривую можно получить при больших объемах выборок $n_0 \geq 250$ [1] значений отдельных измерений, достаточно малой ширине интервалов d см. рис. 3.11. и чтобы в каждый интервал попадало много x_i , что реализуемо при достаточно высокой чувствительности средства измерений.

Представление о характере поведения кривой закона распределения можно получить и при объемах выборки меньше выше обозначенного а зависимости от типа закона распределения.

Эмпирическая оценка плотности вероятности для каждого интервала шириной d $P_i = n_i / n_0 d$ $i = 1 \div m_0$ - это ординаты кривой закона распределения. Абсцисса вычисляется

как $y_i = y_{i-1} + d$ $i = 2 \div m_0$ а $y_1 := x_{\min} + d/2$ где x_{\min} - минимальное значение отдельного измерения в выборке объема n_0 .

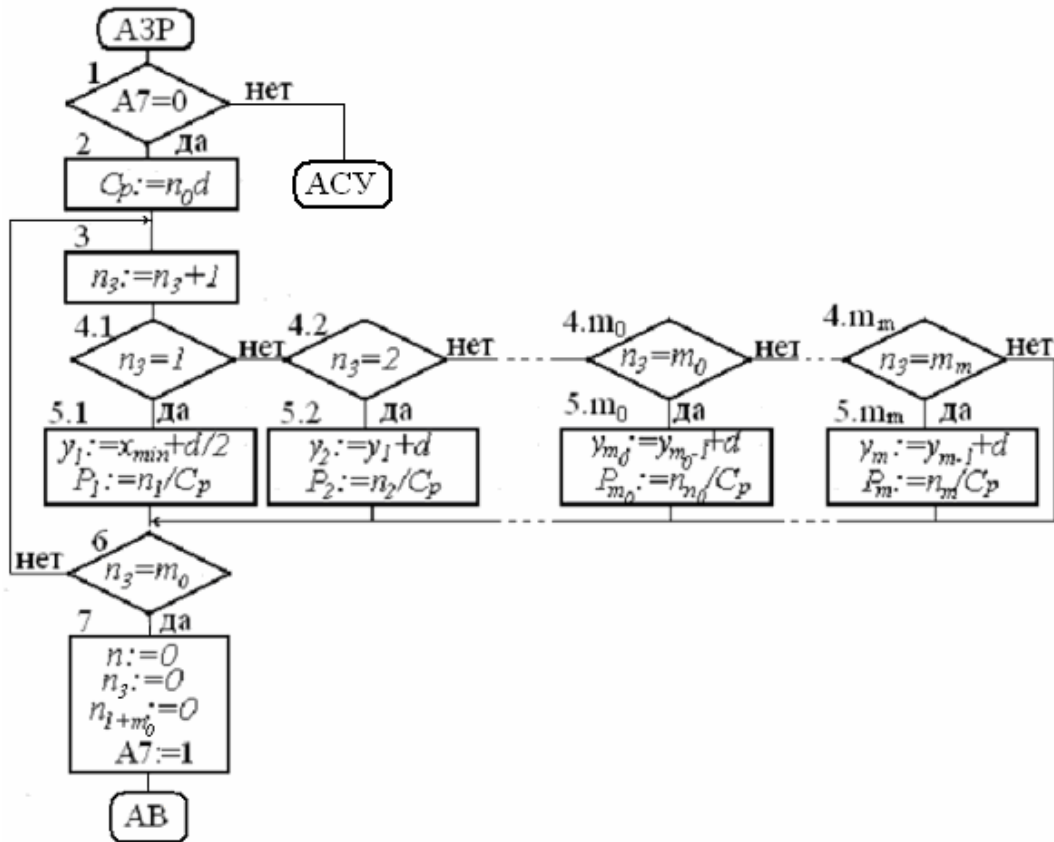


Рис. 3.10

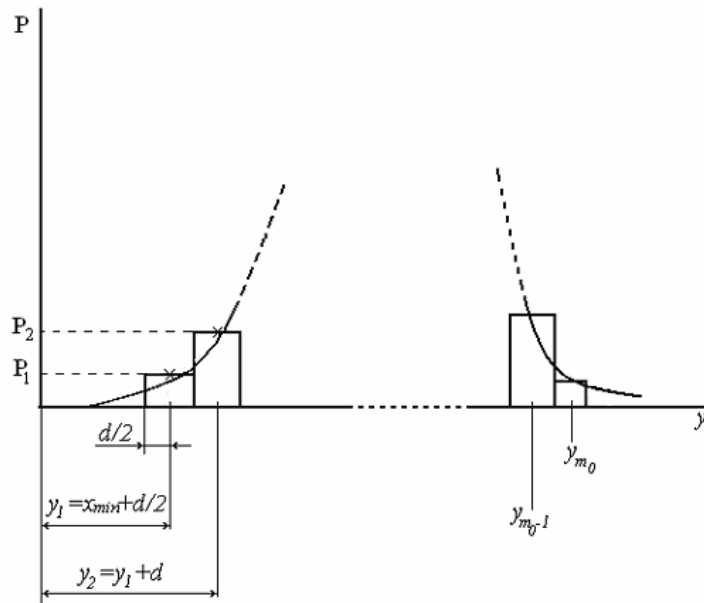


Рис. 3.11

Алгоритм может быть задействован только после АВВ где определяются x_{\min} и АЭЖ где вычисляется ширина интервала d , определяются n_i и выполняется за один заход.

В алгоритме рис. 3.10 выполняются следующие операции. Признаком $A7$ бл.1 по ветви “нет” блокируется выполнение алгоритма с переходом на выполнение следующего алгоритма АСУ, если оператором в начальной настройке этому признаку присваивается единица т.е, если $A7 := 1$. Если $A7 := 0$ то ветви “да” бл.1 вычисляется параметр $C_p := n_0 d$ (см. эмпирическую плотность вероятности) бл.2 и дальше включается счетчик точек на кривой закона распределения $n_3 := n_3 + 1$ бл.3. Блоки 4 опрашивают счетчик n_3 координаты какой точки вычислить и по ветви “да” они вычисляются, что отражено в блоках 5. Если количество просчитанных точек не достигло m_0 т.е. $m_3 < m_0$ бл.6, то по ветви “нет” следует возврат к блоку 3 для вычисления координат очередной точки. Если $n_3 = m_0$ по ветви “да” бл.6 алгоритм приводится в исходное обнулением $n_3 := 0$ бл.7. Кроме того обнуляются счетчики n_i в АЭК $n_{1 \div m_0} := 0$ и формируется признак $A7 := 1$ блокирующий выполнение алгоритма; обнуляется счетчик отдельных измерений в алгоритме “Выборка” $n := 0$.

Таблица переменных параметров АЗР

№ п/п	Наименование параметра	Идентификатор	Диапазон измерения	Дискретность	Где Формируется	Где используется	Начальная настройка	Примечания
1	Признак блокирующий выполнения алгоритма	$A7$	$0 \vee 1$	1	АЗР	АЗР		Нач.наст.-ка задается оператором
2	Переобозначение при вычислении эмпирической плотности вероятности	C_p			АЗР	АЗР	0	
3	Счетчик точек на кривой закона распределений	n_3		1	АЗР	АЗР	0	
4	Оценка координаты кривой закона распределений по оси абсцисс	y_i			АРВ	АЗР	0	$i = 1 \div m_0$ документируется
5	Минимальное значение отдельного измерения в выборке объема n_0	x_{\min}			АЗР	АЗР АЭК	-	
6	Оценка координаты на кривой закона распределения по оси ординат	P_i			АЗР	-	0	$i = 1 \div m_0$ документ-ется
7	Кол-во значений отд. измерений в интервалах шириной d	$n_{i \div m_0}$		1	АЭК	АЗР	-	
8	Ширина интервала на которые разбит размах выборки	d			АЭК	АЗР	-	
9	Счетчик отдельных измерений в АВ	n		1				

Таблица постоянных параметров АЗР

№ п/п	Наименование параметра	Идентификатор	Значения или диапазон реакции	Дискретность	Где используется	Примечания
1	Объем используемой выборки	n_0		1	АА,АРВ,АЭК, АСУ, АЗР	Коррек.-ся оператором
2	Количество вычисленных точек на кривой закона распределения то же, что и количество интервалов шириной d	m_0		1	АЭК,АЗР	Задается оператором в алгоритме АЭК
3	Максимальный объем счетчика n_3	m_m		1	АЗР,АЭК	

3.9. Алгоритм "Статистическая устойчивость" (АСУ).

На момент включения АСУ должны отработать алгоритмы АС, АСКП, АРВ т.е. должны быть вычислены среднее арифметическое выборки \bar{x} , выборочная средняя квадратическая погрешность $\sigma(x_i)$ и размах выборки R . АСУ при его реализации в виде программного модуля должен решать две задачи.

Первая задача – это исследование периферийных областей (краев) реального закона распределения с целью определения коэффициента K_B для доверительного интервала $\bar{x} \pm K_B \sigma(x_i)$, соответствующего выбранной доверительной вероятности P_B . Эта задача решается посредством выяснения зависимости эмпирической доверительной вероятности $P_i = \Delta n_i / n_0$ (Δn_i - количество отдельных измерений попадающих в интервал $\bar{x} \pm K_i \sigma(x_i)$) от коэффициента K_i , используя выборки объема n_0 . Как известно, коэффициенты K представляют из себя отношение $\Delta x / \sigma(x_i)$, где Δx удаленность по оси x от \bar{x} (см. разд. 1.4). Чтобы примерно оценить диапазон изменения K в случае конкретной выборки можно положить $\Delta x = R/2$ и тогда наибольшее значение K будет $K_m = R/2\sigma(x_i)$; наименьшее $K=0$. Нужно также отметить, что K_m случайная величина и может изменяться как для выборок одинакового объема, так тем более для выборок разного объема, а то, что при $K_i = K_m$ $P_i = 1$ является следствием неизбежной ограниченности используемых выборок. Если закон распределения устойчив, то закономерность в поведении кривой $P_i = f(K_i)$ рис. 3.12 проявится и после обработки одной выборки, но достаточно большого объема $n_0 = (100 \div 400)$. Это будет зависеть не только от объема выборки, но и от типа закона распределения. Обработав две три выборки достаточного объема и обнаружив наложение кривых $P_i = f(K_i)$, а также закономерность в их поведении можно достаточно точно определить K_B , соответствующего выбранной доверительной вероятности P_B .

Вторая задача – это контроль статистической устойчивости измерительной информации посредством вычисления, с использованием теперь уже известного K_B , эмпирической доверительной вероятности $P_3 = \Delta n_3 / n_0$, где Δn_3 - количество отдельных измерений попавших в интервал $\bar{x} \pm K_B \sigma(x_i)$, для каждой контролируемой выборки, ее документирования и сравнения с P_B . Условие $P_3 \geq P_B$, как критерий статистической устойчивости измерительной информации может соблюдаться не столько для каждой конкретной выборки, сколько в среднем в процессе обработки контролируемых выборок

т.е. во времени так, например, в случае нормального закона распределения при $P_B = 0,997$ и объеме выборки $n_0 = 370$ за пределами доверительного интервала $\bar{x} \pm K_B \sigma(x_i)$ в среднем т.е. от выборки к выборке может оказаться одно отдельное измерение [3, 5]. Для какой-то выборки это может быть ноль и следовательно $P_3 > P_B$, а для следующей за ней два и следовательно $P_3 < P_B$, а в среднем $P_3 = P_B$ для этих двух выборок. Добиваться строгого соблюдения условия $P_3 > P_B$ посредством увеличения K_B нет смысла, поскольку можно потерять контроль над статистической устойчивостью измерительной информации. Такой подход соответствует т.н. детерминистскому (или идеологии запасов) в результате чего возникают необъяснимые отказы, неизвестное количество брака и т.п.

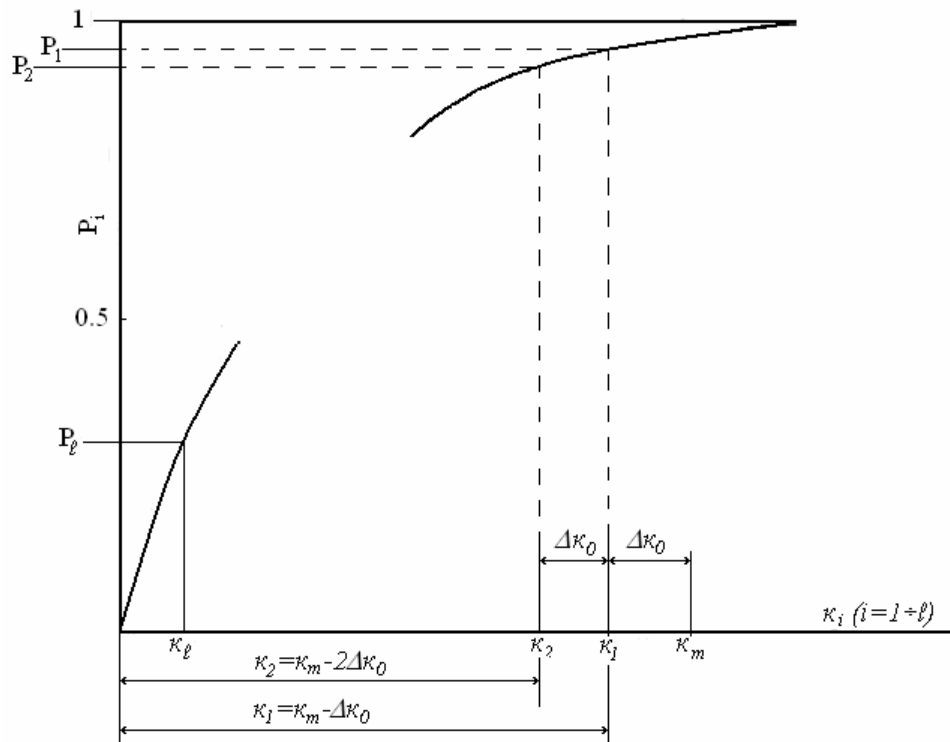


Рис. 3.12

Чтобы получить эмпирическую зависимость $P_i = f(K_i)$ можно с каким-то шагом ΔK_0 , начиная от наибольшего K_m вычислять отступления от него $\Delta K = N \Delta K_0$ (где N - счетчик количества шагов или точек на кривой $P_i = f(K_i)$), после чего вычислять очередное значение $K_i = K_m - \Delta K$. И дальше, вычисляя отступление от \bar{x} $\Delta x = K_i \sigma(x_i)$ для каждого K_i , вычислять границы доверительных интервалов $x_M = \bar{x} - \Delta x$, и $x_B = \bar{x} + \Delta x$, а посредством проверки $x_M \leq x_i \leq x_B$ (x_i - значения отдельных измерений в выборке объема n_0) подсчитать Δn_i - количество попадающих в соответствующие доверительные интервалы x_i . В итоге вычислять $P_i = \Delta n_i / n_0$ для каждого интервала и построить зависимость $P_i = f(K_i)$ рис. 3.12.

Проверка каждого x_i на попадание в соответствующие доверительные интервалы должна начинаться с наибольшего, который последовательно уменьшается с каждым шагом и при условии попадания в соответствующих счетчиках $\Delta n_i := \Delta n_i + 1$

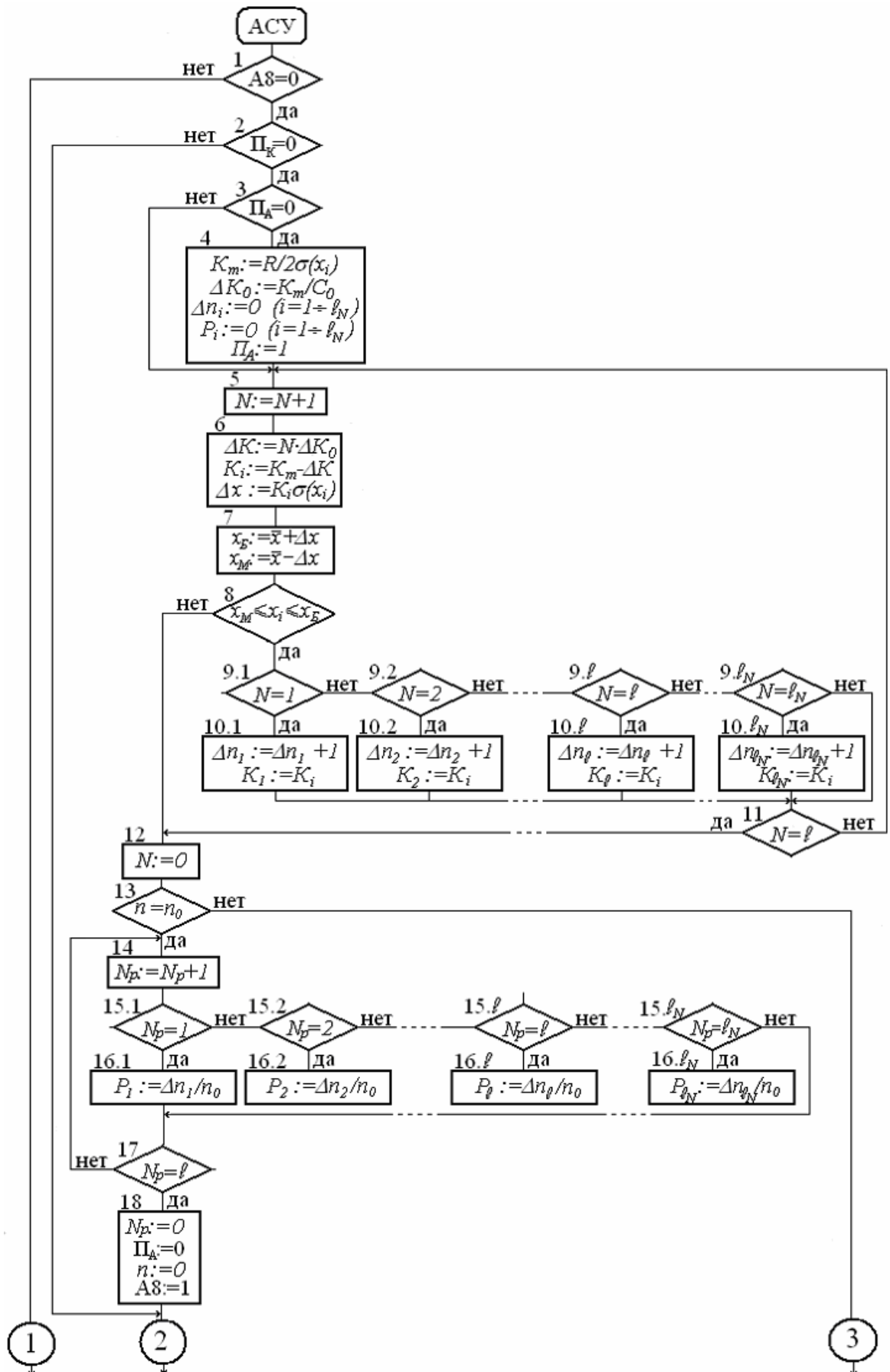
прибавляется единица. Если же в очередной меньший доверительный интервал x_i не попадает, ясно, что дальше продолжать проверку нет смысла, а нужно начинать проверку сначала с очередным x_i и так последовательно проверить всю выборку объемом n_0 .

Организация последовательности выполнения выше обозначенных операций отражена в структурной блок-схеме алгоритма рис. 3.13. Признак A_8 бл.1 блокирует выполнение алгоритма при $A_8 = 1$ или разрешает его выполнение при $A_8 = 0$ по ветви “да”. Признак Π_k бл.2 блокирует выполнение первой задачи по определению K_B при $\Pi_k = 1$ или разрешает выполнение этой задачи при $\Pi_k = 0$ по ветви “да”. Значения признаков A_8 и Π_k задаются оператором в начальной настройке параметров алгоритма. Признак Π_A бл.3 разрешает разовые вычисления в бл.4 на первом заходе в алгоритм при $\Pi_A = 0$ по ветви “да” и блокирует их при последующих заходах при $\Pi_A = 1$. Присвоение $\Pi_A := 1$ производится в бл.4 на первом заходе в алгоритм. В бл.4 отражено вычисление $K_m = R / 2\sigma(x_i)$ и шаг (расстояние между точками по оси K) $\Delta K_0 = K_m / C_0$ с которым будут вычисляться K_i и P_i . C_0 число, значение которого задает оператор. При этом нужно учитывать то, что счетчик точек $N := N + 1$, бл.5 в диапазоне изменения $K = 0 \div K_m$ ограничен своим максимальным значением ℓ_N . Если $C_0 > \ell_N$, то счетчик может быть заполнен до конца, что в общем-то необязательно и при исследовании периферийных областей закона распределения можно ограничиться $\ell = 3 \div 5$ точек. Если $C_0 \leq \ell_N$ счетчик точек $N := N + 1$ бл.5 должен заполняться не больше, чем до ℓ равной целой части разности $C_0 - 1$, поскольку следующий доверительный интервал может быть равным нулю.

В бл.6 отражено вычисление $\Delta K = N \cdot \Delta K_0$ т.е. отступление от K_m в виде количества шагов задаваемого счетчиком $N := N + 1$ бл.5. В том же бл.6 также отражены вычисления очередных значений $K_i := K_m - \Delta K$ и отступления от среднего \bar{x} $\Delta x := K_i \sigma(x_i)$.

В бл.7 отражено вычисление границ очередного доверительного интервала $x_B := \bar{x} + \Delta x$ и $x_M := \bar{x} - \Delta x$.

В бл.8 отражена проверка значений отдельных измерений x_i на попадание в доверительные интервалы $x_M \leq x_i \leq x_B$. Последовательность проверок задается счетчиком $N := N + 1$ бл.5 и начинается с наибольшего интервала при $N = 1$. Если x_i попадает в наибольший интервал то по ветви “да” бл.8 и по ветви “да” блок 9.1 при $N = 1$ счетчик $\Delta n_1 := \Delta n_1 + 1$ в бл.10.1 прибавляет единицу и также фиксируется соответствующее значение $K_1 := K_i$. Далее по этой же ветви следует бл.11 где контролируется количество заданных оператором точек ℓ и если $N < \ell$, то следует возврат к счетчику $N := N + 1$ бл.5 для проверки того же x_i на попадание в меньший интервал с меньшим K_i . при $N = 2$. Этот цикл с пошаговым уменьшением доверительных интервалов продолжается до тех пор пока проверка в бл.8 покажет непопадание в какой-то очередной доверительный интервал, тогда по ветви “нет” бл.8 обнуляется счетчик $N = 0$ бл.12 и если выборка до конца еще не использована то по ветви “нет” бл.13 следует возврат к алгоритму “Выборка” (АВ) за очередным x_i . Если выборка объемом n_0 использована до конца, тогда (при $n = n_0$) по ветви “да” бл.13 должны быть вычислены все P_i соответствующие K_i . С этой целью включается счетчик $N_p := N_p + 1$ бл.14 и при $N_p = 1$ по ветви “да” бл.15.1 вычисляется $P_1 = \Delta n_1 / n_0$ и дальше по этой ветви следует бл.17, который контролирует количество вычисляемых P_i задаваемое оператором.



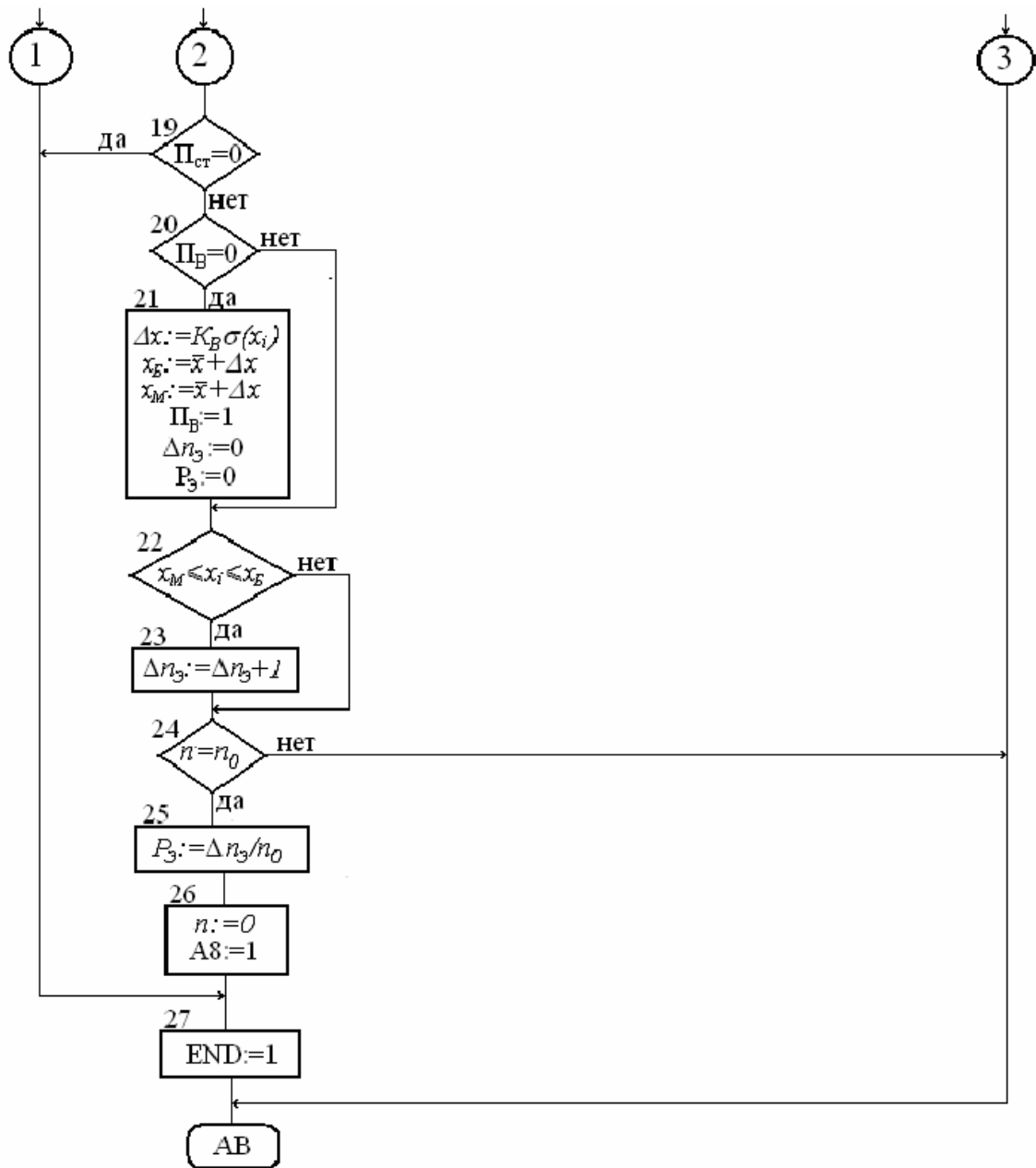


Рис. 3.13.

Если $N_p < \ell$ то по ветви “нет”, бл.17 следует возврат к счетчику $N_p := N_p + 1$ бл.14. При $N_p = 2$ вычисляется P_2 и т.д. до $P_\ell := \Delta n_\ell / n_0$. При $N_p = \ell$ бл.17 по ветви “да” обнуляются параметры в бл.18 $N_p := 0$, $\Pi_A := 0$, $n := 0$, чем АСУ и АВ приводятся в исходное и там же производятся присвоения $A8 := 1$ чем блокируется выполнение алгоритма.

По полученным данным строится зависимость $P_i = f(K_i)$, примерный вид которой изображен на рис. 3.12. При повторных запусках алгоритма подбирая C_0 можно изменить шаг между точками, добиваясь оптимального вида эмпирической зависимости $P_i = f(K_i)$, особенно в периферийной области реального закона распределения. Исследовав 2-3 выборки можно определить K_B для выбранного P_B .

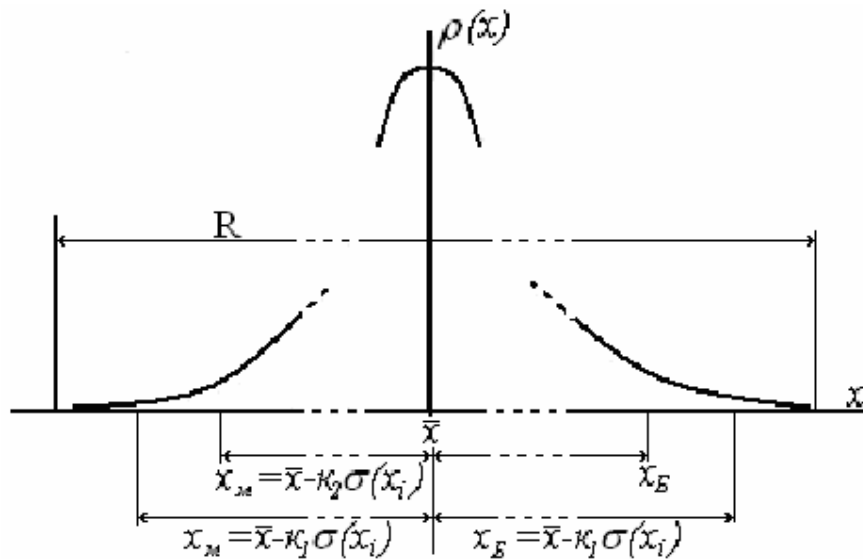


Рис. 3.14

Вторая часть алгоритма по контролю статистической устойчивости измерительной информации запускается при $A8 = 0$ по ветви “да” бл.1, при $\Pi_k = 1$ по ветви “нет” бл.2, при $\Pi_{CT} = 1$ по ветви “нет” бл.19 - эти параметры задаются оператором в начальной настройке переменных параметров алгоритма. При $\Pi_B = 0$ бл.20 по ветви “да” на первом заходе в алгоритм должны быть выполнены разовые операции обозначенные в бл.21 - это обнуление счетчика отдельных измерений $\Delta n_3 := 0$, попадающие в доверительный интервал $\bar{x} \pm K_B \sigma(x_i)$ соответствующий выбранной доверительной вероятности P_B ; обнуление эмпирической доверительной вероятности $P_3 := 0$, вычисляемой при обработке каждой контролируемой выборки. В бл.21 отражено также вычисление отступления от \bar{x} с использованием уже известного K_B т.е. $\Delta x := K_B \sigma(x_i)$ и границ доверительного интервала $x_M := \bar{x} - \Delta x$ и $x_B := \bar{x} + \Delta x$. Там же в бл.21 отражено присвоение $\Pi_B := 1$, чем блокируются выше перечисленные разовые операции в бл.21 по ветви “нет” бл.20.

В бл.22 отражена проверка попадания значения x_i в интервал $\bar{x} \pm \Delta x$ и если оно попадает в этот интервал, то тогда по ветви “да” в счетчике $\Delta n_3 := \Delta n_3 + 1$ бл.23 прибавляется единица, если же x_i не попадает в интервал $\bar{x} \pm \Delta x$, то по ветви “нет” счетчик в бл.23 обходится.

Если выборка не использована до конца т.е. $n < n_0$ то по ветви “нет” бл.24 происходит возврат к AB за очередным x_i . Если же $n = n_0$ то по ветви “да” бл.24 в бл.25 отражено вычисление эмпирической доверительной вероятности $P \varepsilon := \Delta n_3 / n_0$ для случая конкретной контролируемой выборки, которая должна документироваться. В бл.26 обнуляется счетчик отдельных измерений в AB $n := 0$ и формируется признак $A8 := 1$ блокирующий выполнение алгоритма и признак выхода из программы $END := 1$.

Таблица переменных параметров АСУ.

№ п/п	Наименование параметра	Идентификатор	Диапазон изменения	Дискретность	Где формируется	Где используется	Настройка	Примечания
1	2	3	4	5	6	7	8	9
1	Признак блокирования выполнения АСУ	А8	0∨1	1	АСУ	АСУ		Нач. настройку задает оператор
2	Признак блокирования первой задачи	ПК	0∨1	1	АСУ	АСУ		
3	Признак блокирования разовых вычислений в бл.4	ПА	0∨1	1	АСУ	АСУ	0	
4	Коэффициент, вычисляемый по полуразмаху выборки	K_m			АСУ	АСУ	0	
5	Размах выборки объемом n_0	R			АРВ	АСУ,АЭК	-	
6	Выборочная средняя квадратичная погрешность	$\sigma(x_i)$			АСКП	АСУ, АА,АК	-	
7	Шаг (расстояние между точками по оси К) в зависимости $P_i=f(K_i)$	ΔK_0			АСУ	АСУ	0	
8	Счетчики отдельных измерений попадающих в доверительные интервалы $\bar{x} \pm K_i \sigma(x_i)$	Δn_i $i = 1 \div \ell$		1	АСУ	АСУ	0	
9	Эмпирические доверительные вероятности соответствующие K_i	P_i $i = 1 \div \ell$			АСУ	АСУ	0	Документируется
10	Коэффициенты при $\sigma(x_i)$ при вычислении границ доверительных интервалов	K_i $i = 1 \div \ell$			АСУ	АСУ	0	
11	Отступление от K_m по оси К к следующей точке на кривой $P_i := f(K_i)$	ΔK			АСУ	АСУ	0	
12	Счетчик точек по кривой $P_i := f(K_i)$ или испытуемых доверительных интервалов	N	$0 \div \ell_N$	1	АСУ	АСУ	0	
13	Среднее арифметическое выборки объемом n_0	\bar{x}			АС	АСУ,АСКП, АА, АК,АЭК	-	

1	2	3	4	5	6	7	8	9
14	Отступление от \bar{x} по оси абсцисс, определяющее границы доверит. интервалов	Δx			АСУ	АСУ	0	
15	Граница дов-ных интервалов - большая	x_B			АСУ	АСУ	0	
16	Граница дов-ных интервалов - меньшая	x_M			АСУ	АСУ	0	
17	Значения отдельных измерений	x_i			АВ	АС,АСКП, АА,АК,АРВ, АЭК,АСУ		
18	Счетчик отдельных измерений в АВ	n		1	АВ	АС,АСКП, АА,АК,АРВ, АЭК,АСУ		
19	Счетчик вычисляемых P_i .	N_p	$0 \div \ell_N$	1	АСУ	АСУ	0	
20	Признак блокирования второй задачи	$П_{ст}$	$0 \vee 1$	1	АСУ	АСУ		Нач.настройку задает оператор
21	Признак блокирования разовых вычислений в бл.21	$П_B$	$0 \vee 1$	1	АСУ	АСУ	0	
22	Счетчик отдельных измерений x_i попадающих в дов-ные интервалы $\bar{x} \pm K_B \sigma(x_i)$	Δn_3		1	АСУ	АСУ	0	
23	Контролируемая эмпирическая доверительная вероятность	P_3			АСУ	АСУ	0	
24	Признак выхода из программы	END	$0 \vee 1$	1	АСУ	АВ	0	

Таблица постоянных параметров АСУ

№ п/п	Наименование параметра	Идентификатор	Значение (Диапазон корреляции)	Дискретность	Где используется	Примечание
1	2	3	4	5	6	7
1	Параметр, определяющий шаг вычисления K_i и P_i	C_0			АСУ	Выбирается оператором
2	Объем счетчика точек на кривой зависимости $P_i = f(K_i)$	ℓ_N		1	АСУ	

1	2	3	4	5	6	7
3	Количество задаваемых оператором точек на кривой $P_i = f(K_i)$	ℓ		1	АСУ	$\ell \leq \ell_n$
4	Коэффициент доверительного интервала соответствующего выбранной доверительной вероятности P_B	K_B			АСУ	Определяется по выполнению первой задачи АСУ
5	Объем используемой выборки	n_0		1	АА,АЭК, АРВ,АСУ	Корректируется оператором

4. Закон распределения Пуассона. Счетчики элементарных частиц.

Результаты счета частиц при постоянной средней плотности их потока (типичный случай в атомной и ядерной физике) подчиняются закону распределения Пуассона [5]. Если счетчик за промежутки времени $\Delta t = const$ регистрирует в каждом из K этих Δt какое-то число частиц

$$n_1, n_2, n_3, \dots, n_k,$$

то эти числа и подчиняются распределению Пуассона.

Среднее число регистрируемых за Δt частиц вычисляется как

$$a = \frac{\sum_{i=1}^k n_i}{K}.$$

Средняя квадратичная погрешность счета частиц будет равна [5]

$$\sigma(n_i) = (a)^{1/2}$$

Вероятность попадания какого-то количества частиц n в интервал времени Δt дается законом распределения Пуассона [5].

$$\omega_a(n) = e^{-a} \frac{a^n}{n!}$$

4.1. Алгоритм “Число частиц” (АЧЧ).

Числа частиц в интервалах времени $\Delta t = const$ $n_1, n_2, n_3, \dots, n_k$ представляют из себя выборку объема K_0 и заводится в базу данных (БД) с экрана монитора ПЭВМ.

Назначение алгоритма рис. 4.1 при каждом заходе в него – поочередное, обеспечиваемое счетчиком $K := K + 1$ бл.2 по ветви “да” блоки 3, переприсвоение параметру n_i значений числа частиц начиная с первого в выборке n_1 до n_{k_0} содержащихся в БД. Эта операция отражена в блоках 4. $K_0 \leq K_m$ где K_m предельно возможный объем выборки. K_0 – задается оператором.

Программа в целом запускается параметром $END = 0$ по ветви “да” бл.1. После выполнения программы в алгоритме “Закон распределения Пуассона” параметру $END := 1$, после чего следует выход из программы по ветви “нет” бл.1.

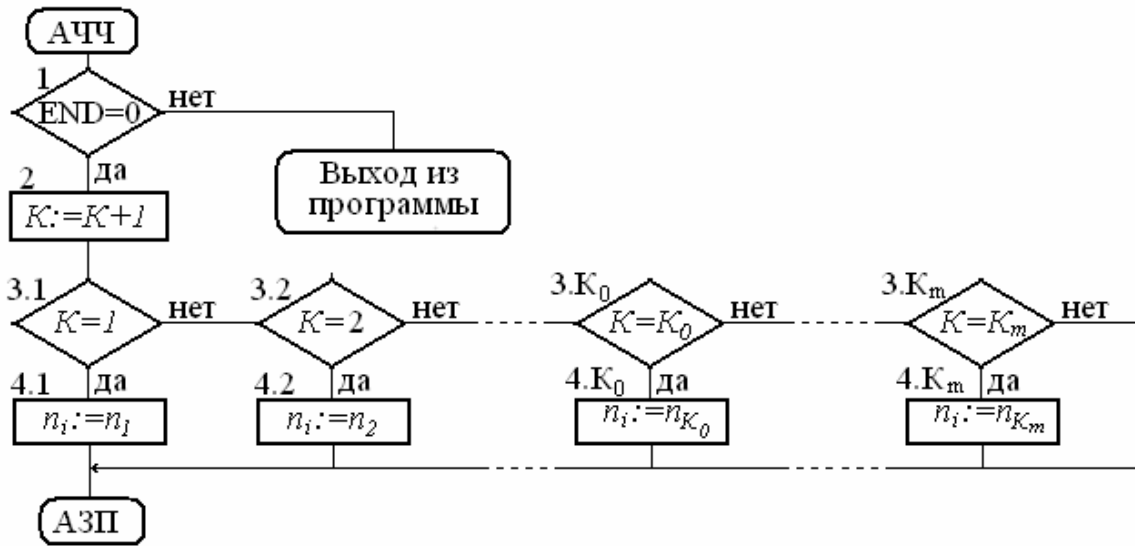


Рис. 4.1.

Блок-схема взаимодействия настоящего алгоритма с алгоритмом АЗП изображена на рис.4.2.



Рис. 4.2.

Блок-схема отражает циклическое взаимодействие двух алгоритмов включающее в себя операции переприсвоения в АЧЧ параметру n_i очередного значения из выборки содержащийся в БД, использование его в АЗП и возврат в АЧЧ к операции переприсвоения за очередным n_i .

Таблица переменных параметров АЧЧ.

№ п/п	Наименование параметра	Идентификатор	Диапазон изменения	Дискретность	Где формируется	Где используется	Начальная настройка	Примечание
1	Признак завершения программы и выхода из нее	END	$0 \vee 1$	1	АЗП	АЧЧ	0	
2	Счетчик интервалов $\Delta t = const$	K		1	АЧЧ	АЧЧ, АЗП	0	
3	Количества частиц попавших в интервалы $\Delta t = const$	$n_{i \rightarrow K_0}$		1	БД	АЧЧ	-	
4	Значение количества частиц попавших в интервалы $\Delta t = const$ после переприсвоения	n_i		1	АЧЧ	АЗП	0	

Таблица постоянных параметров АЧЧ

№ п/п	Наименование параметра	Идентификатор	Значение или диапазон корреляции	Дискретность	Где используется	Примечание
1	Объем счетчика интервалов Δt равный возможному объему выборки в БД	K_m		1	АЧЧ	
2	Объем используемой выборки	K_0		1	АЧЧ	$K_0 \leq K_m$

4.2. Алгоритм “Закон распределения Пуассона” (АЗП)

В АЗП должны вычисляться среднее число регистрируемых частиц a за $\Delta t = const$, средняя квадратичная погрешность σ , вероятность попадания n частиц от 1 до n_0 в интервал времени Δt т.е. выясняется зависимость $\omega_\alpha(n)$.

В бл.1 алгоритма рис. 4.3. отражена операция суммирования числа частиц n_i , т.е. $\sum n_i := \sum n_i + n_i$. Если число суммированных n_i $K < K_0$ то по ветви “нет” бл.2 следует возврат к АЧЧ за очередным n_i . Если $K = K_0$ то по ветви “да” следуют вычисления a бл.3, σ бл.4 и $E = e^{-a}$ бл.5.

В алгоритме реализован вариант относительно низкой средней плотности потока частиц, когда могут интересоваться вероятности попадания частиц в интервалы Δt от одной и больше.

Счетчик $n := n + 1$ бл.6 последовательно задает число частиц от одной до n_0 вероятность попадания которых в интервал Δt вычисляется по ветви “да” блоки 7. Вычисление $\omega(n)$ отражено в блоках 8.

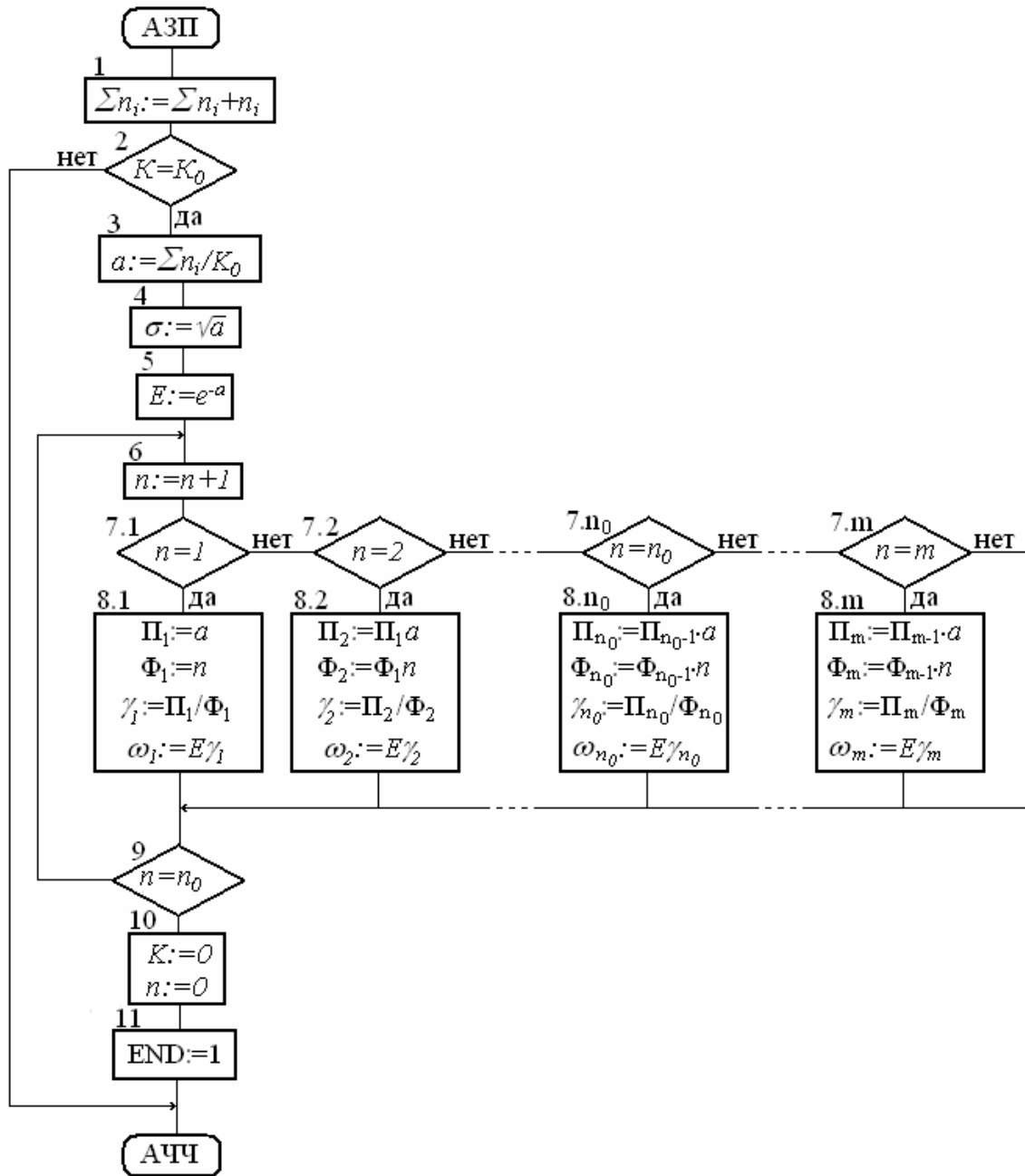


Рис. 4.3.

a^n представляет из себя произведение a само на себя $n_{раз}$. При возрастании n через единицу проще всего a^n вычислить как произведение предыдущего значения на a т.е. $\Pi_1 := a$, $\Pi_2 := \Pi_1 a$ и т.д. т.е. $\Pi_n := \Pi_{n-1} \cdot a$, что и отражено в блоках 8. Тот же прием можно использовать и при вычислении $n!$, используя значения счетчика $n := n + 1$. Если на счетчике $n = 1$, то $n! = \varphi_1 = n = 1$, если $n = 2$, $n! = \varphi_2 = \varphi_1 \cdot n = 1 \cdot 2 = 2$, если $n = 3$, $n! = \varphi_3 = \varphi_2 n = 2 \cdot 3 = 6$ и т.д. т.е. по принципу $\varphi_n = \varphi_{n-1} \cdot n$. Эти операции тоже отражены в

блоках 8. Там же отражены и вычисления соответствующих частных $\gamma_n = a^n / n! = \Pi_n / \Phi_n$ (см. формулу для $\omega(n)$), и вероятностей ω_{1+m} попадания $n \geq 1$ частиц в интервалы $\Delta t = const$. После завершения вычислений ω_{1+m} и соответствующее число частиц n должны документироваться в отдельном файле с возможностью вывода его содержания на экран ПЭВМ. Обработка программных модулей потребует конкретных данных с использованием которых должен рассчитываться контрольный пример.

Таблица переменных параметров АЗП

№ п/п	Наименование параметра	Идентификатор	Диапазон изменения	Дискретность	Где формируется	Где используется	Начальная настройка	Примечание
1	Значения количества частиц, попавших в интервалы $\Delta t = const$ после переприсвоения.	n_i		1	АЧЧ	АЗП	-	
2	Сумма n_i	$\sum n_i$		1	АЗП	АЗП	0	
3	Среднее число регистрируемых частиц за интервалы $\Delta t = const$	a			АЗП	АЗП	0	Документируется
4	Средняя квадратичная погрешность счета частиц	σ			АЗП		0	Документируется
5	Промежуточный параметр при вычислении $\omega_a(n)$	E			АЗП	АЗП	0	
6	Счетчик интервалов времени $\Delta t = const$ или количества n_i	K		1	АЧЧ	АЗП	-	
7	Счетчик задающий число частиц вероятности попадания которых в интервалы $\Delta t = const$ вычисляется	n		1	АЗП	АЗП	0	Документируется
8	Произведения: $\Pi_{n-1} \cdot a = \Pi_n = a^n$	Π_{1-m}			АЗП	АЗП	0	
9	Произведения: $\Phi_{m-1} n = \Phi_n = n!$	Φ_{1+m}			АЗП	АЗП	0	
10	Частные: $a^n / n!$	γ_{1+m}			АЗП	АЗП	0	
11	Вероятности попадания n частиц в интервалы $\Delta t = const$	ω_{1+m}	$0 \div 1$		АЗП		0	Документируется
12	Признак завершения программы и выхода из нее	END	$0 \vee 1$	1	АЗП	АЧЧ	0	

Таблица постоянных параметров АЗП

№ п/п	Наименование параметра	Идентификатор	Значение или диапазон коррекции	Дискретность	Где используется	Примечание
1	Количество усредняемых n_i	K_0		1	АЗП	Корректируется оператором
2	Наибольшее число частиц на конкретный момент для которого вычисления $\omega(n)$	n_0		1	АЗП	Корректируется оператором
3	Объем счетчика $n := n + 1$	m		1	АЗП	$n_0 \leq m$

5. Алгоритм фильтрации сбоев “Срединное” (АФС).

Причины возникновения сбойных ситуаций в измерительной информации средств измерений СУ весьма неопределенны – это и взаимовлияние различных устройств и приборов одновременно работающих на объекте и возможное влияние кратковременно изменившихся внешних условий и чисто технические причины связанные с конкретным прибором, когда нужно блокировать использование измерительной информации с этого прибора. Поэтому фильтрация измерительной информации имеет свой смысл.

Алгоритм “Срединное” может быть реализован в вычислительном комплексе СУ с целью фильтрации сбойных значений отдельных измерений. К его достоинствам можно отнести и то, что улучшается точность измерительной информации, поскольку из трех (или из пяти) для использования выбирается срединное между большим и меньшим значениями отдельных измерений.

Безусловно, что такие параметры сбойных ситуаций как момент возникновения или длительность непредсказуемы. Если в стационарных условиях измерений момент сбоя можно уловить и парировать посредством непрерывного сравнения значений отдельных измерений и выбора срединного из трех, а также контроля размаха такой выборки с использованием параметра точности прибора, то длительность сбойной ситуации можно парировать только на уровне управления объектом, контролируя его состояние в целом. Непредсказуемы также и значения сбойных отдельных измерений, они могут не намного выходить за рамки параметра точности, что можно принять за эволюцию входного сигнала прибора, поэтому выбор срединного существенно увеличивает надежность измерительной информации.

Признак Π_ϕ в алгоритме рис. 5.1 формируется программой управления объектом и на время эволюции входного сигнала прибора должен блокировать работу алгоритма по ветви “нет” бл.1 при $\Pi_\phi = 1$. При $\Pi_\phi = 0$ алгоритм запускается по ветви “да” бл.1.

Алгоритм настроен для тактированного режима работы вычислительного комплекса, когда измерительная информация на входных регистрах обновляется один раз за такт работы комплекса. Поскольку длительность такта может быть 50, 100, 200 мс, то сбойные значения отдельных измерений могут “залипать” т.е. прибор за время такта может не среаксировать или длительность сбойной ситуации сравнима с несколькими тактами работы вычислительного комплекса. Эту ситуацию можно парировать двумя способами – регистрировать значения отдельных измерений через несколько тактов т.е. прореживать и полностью обновлять выборку из $K = 3$ при получении первого сбоя. Задачу

прореживания решают блоки 2, 3, 4. Счетчик тактов $c := c + 1$ бл.2 отсчитывает число тактов которые нужно пропустить и при $c \leq t$ бл.3 (где t – число пропускаемых тактов) по ветви “да” алгоритм блокируется. При $c > t$ по ветви “нет” бл.3 обнуляется счетчик $c := 0$ бл.4 и срабатывает алгоритм после чего снова будет пропущено t тактов.

В течение первых трех срабатываний алгоритма значения отдельных измерений считываются с входного регистра вычислительного комплекса и запоминаются для их последующего использования. Эта процедура отражена в бл.5. Всего значений три N_1, N_2, N_3 . N_1 значение отдельного измерения зарегистрированное на такте срабатывания алгоритма (т.е. на настоящем такте), N_2 на предыдущем по отношению к N_1 , N_3 – на предыдущем по отношению к N_2 т.е. за три последовательных срабатывания алгоритм запоминает три значения посредством переприсвоения $N_3 := N_2, N_2 := N_1$ и $N_1 := N$ где N значение отдельного измерения на входном регистре вычислительного комплекса в настоящем такте. Процедура переприсвоения отраженная в бл.5 во всех последующих срабатываниях алгоритма является и процедурой обновления измерительной информации. N_3 как наиболее “старое” исключается посредством переприсвоения ему N_2 , дальше N_2 переприсваивается значение N_1 , а N_1 присваивается новое значение с входного регистра при каждом заходе в алгоритм. По ветви “нет” блок 6 на стадии регистрации первых трех значений отдельных измерений при $i \leq K$ ($K = 3$) остальная часть алгоритма блокируется. Счетчик $i := i + 1$ бл.7 подсчитывает число запоминаемых значений отдельных измерений. При $i = 3$ по ветви “да” бл.6 включается процедура нахождения срединного посредством обнаружения наибольшего и наименьшего из трех значений отдельных измерений блоки 8 ÷ 15.

Первая операция в этой процедуре это сравнение N_1 с N_2 , бл.8. Если $N_1 \leq N_2$ то по ветви “да” значение меньшего из них присваивается $x_1 := N_1$ а большего $x_2 := N_2$ бл.9, если же $N_1 > N_2$ то по ветви “нет”, бл.8 наоборот значения меньшего присваивается N_2 т.е. $x_1 := N_2$ а большего $x_2 := N_1$. Дальше нужно выяснить положение на числовой оси значения N_3 . Если $N_3 \leq x_1$ то это будет означать, что N_3 меньше меньшего и по ветви “да” бл.11 последуют переприсвоения бл.13 наибольшему x_3 присваивается x_2 т.е. $x_3 := x_2$, срединному $x_2 := x_1$ и наименьшему $x_1 := N_3$. Если $N_3 > x_1$ то по ветви “нет”, бл.11 сравнивается N_3 с x_2 бл.12. Если $N_3 \leq x_2$ то по ветви “да” бл.12 последуют переприсвоения наибольшему $x_3 := x_2$, а срединному $x_2 := N_3$, x_1 остается прежним бл.14. И наконец, третья ситуация если $N_3 > x_2$ то по ветви “нет” бл.12 наибольшим становится N_3 т.е. $x_3 := N_3$ бл.15, x_1 и x_2 остаются прежними. Таким образом, срединное во всех возможных ситуациях будет найдено.

Поскольку процедура обновления информации т.е. исключения наиболее “старого” N_3 и регистрация нового значения $N_1 := N$ и процедура нахождения срединного повторяются в каждом такте срабатывания алгоритма, то поступившее в очередном такте срабатывания алгоритма сбойное значение отдельного измерения будет или наибольшим x_3 или наименьшим x_1 и в процессе обновления измерительной информации оно будет пропущено, а для возможного использования должно быть взято только x_2 . Описанной процедурой сам факт возможного единичного сбоя не фиксируется.

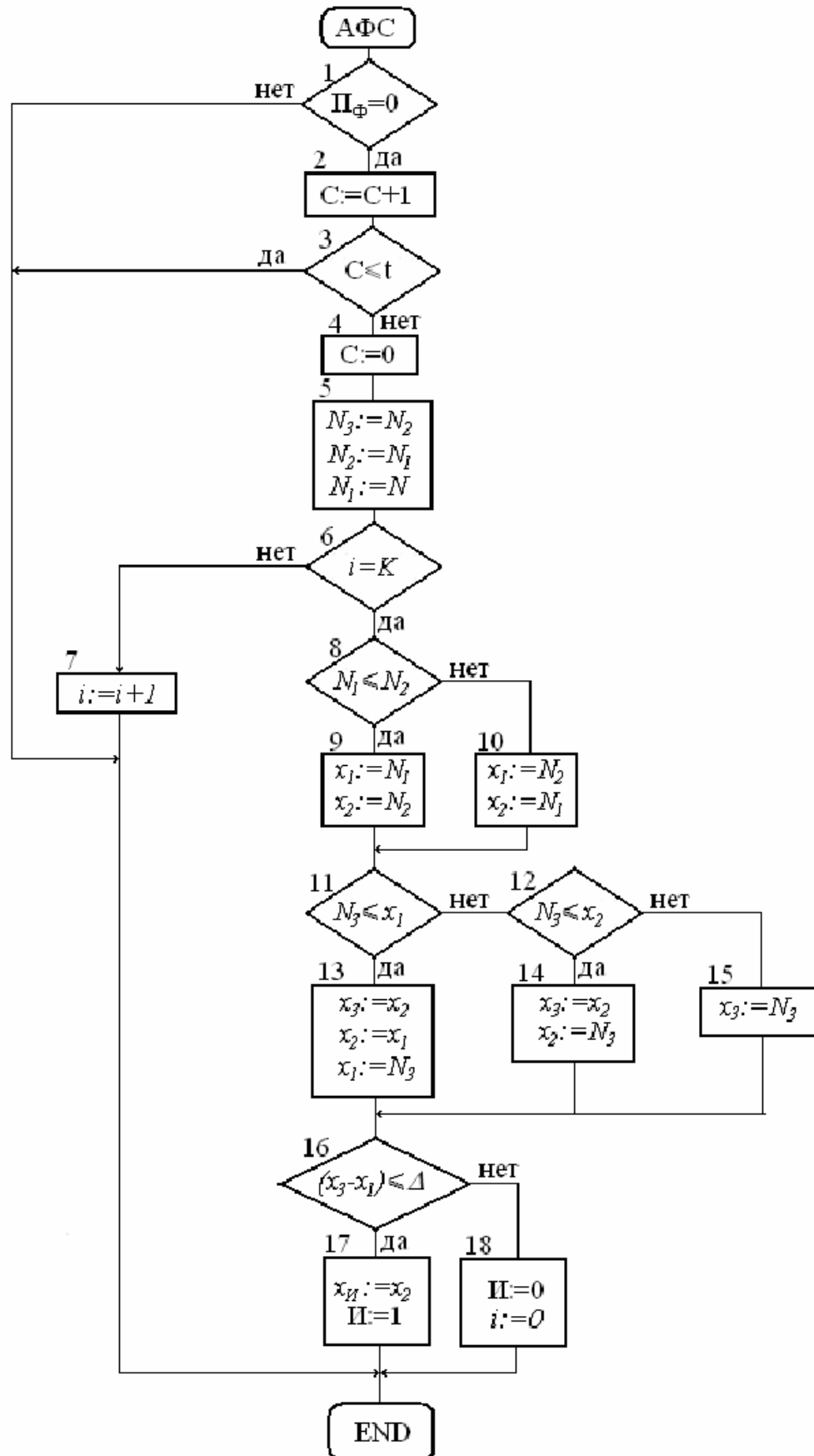


Рис.5.1.

Учитывая непредсказуемость сбойных ситуаций по моменту появления, длительности, значению отдельного измерения все же фиксация момента появления сбоя может оказаться весьма полезной имея в виду выше упомянутые возможные “залипания” обусловленные длительностью сбойной ситуации большей, чем время релаксации средства измерения. Время релаксации может быть учтено параметром t бл.3. т.е. количеством пропускаемых “холостых” тактов отсчитываемых счетчиком $c := c + 1$ бл.7. Чтобы зафиксировать момент появления первого сбойного значения контролируется размах выборки $(x_3 - x_1) \leq \Delta$ бл.16 где Δ характеристика точности прибора.

Допустим, что закон распределения погрешностей нормальный, а выбранная доверительная вероятность 0,997. Это означает, что разности любых двух значений отдельных измерений не должны превышать величины $\Delta = 6\sigma$ с той же вероятностью, поскольку все их значения должны лежать в интервале $\bar{x} - 3\sigma \leq x_i \leq \bar{x} + 3\sigma$ где x_i любые из значений отдельных измерений, \bar{x} среднее арифметическое вычисленное по большому объему выборки, $\sigma(x_i)$ - средняя квадратичная погрешность. В других случаях Δ определяется по достаточному количеству испытаний и задается разработчиком прибора.

В отсутствие сбоя должно выполняться условие $(x_3 - x_1) \leq \Delta$, поэтому по ветви “да” бл.16 формируется значение отдельного измерения для его использования в бл.17 $x_{II} := x_2$, а также формируется признак информативности прибора $I := 1$. Если $(x_3 - x_1) > \Delta$, то это означает, что или x_3 или x_1 сбойные, поэтому по ветви “нет” бл.16 обнуляется признак информативности прибора $I := 0$, а также обнуляется счетчик $i := 0$ бл.18 с целью обновления всех трех значений отдельных измерений.

Ясно, однако, что алгоритм может быть приспособлен к любому режиму работы вычислительного комплекса в том числе и к не тактированному, посредством отсчета времени между регистрацией значений отдельных измерений.

Таблица переменных параметров А.Ф.С

№ п/п	Наименование параметра	Идентификатор	Диапазон изменения	Дискретность	Где формируется	Где используется	Начальная настройка	Примечание
1	2	3	4	5	6	7	8	9
1	Признак блокирующий работу алгоритма	P_ϕ	$0 \vee 1$	1	АФС	Алгоритмы управления объектом	-	
2	Счетчик холостых тактов	c	$0 \div 4$	1	АФС	АФС	0	
3	Значения отдельного измерения – первое в порядке поступления	N_3			АФС	АФС	0	
4	Значения отдельного измерения – второе в порядке поступления	N_2			АФС	АФС	0	
5	Значение отдельного измерения – третье в порядке поступления	N_1			АФС	АФС	0	

1	2	3	4	5	6	7	8	9
6	Значения отдельного измерения на входном регистре вычисленного комплекса	N			Алгоритмы обмена	$A\Phi C$	-	
7	Счетчик отдельных измерений задействованных в алгоритме	i	$0 \div 3$	1	$A\Phi C$	$A\Phi C$	0	
8	Наименьшее значение отдельного измерения из трех используемых	x_1			$A\Phi C$	$A\Phi C$	0	
9	Срединное значение выборки из объема $K = 3$	x_2			$A\Phi C$		0	Документируется
10	Наибольшее значение отдельного измерения из трех используемых	x_3			$A\Phi C$	$A\Phi C$	0	
11	Используемое значение отдельного измерения	x_{II}			$A\Phi C$	Алгоритмы управления объектом	0	Документируется
12	Признак информативности измерительного прибора	I	$0 \vee 1$	1	$A\Phi C$	Алгоритмы управления объектом	0	

Таблица постоянных параметров АФС

№ п/п	Наименование параметра	Идентификатор	Значения или диапазон коррекции	Дискретность	Где используется	Примечание
1	Количество холостых тактов	t		1	$A\Phi C$	Возможная коррекция
2	Количество используемых значений отдельных измерений	K	3	1	$A\Phi C$	
3	Параметр точности средства измерений	Δ			$A\Phi C$	

6. О центрах разработки программного обеспечения для ПЭВМ

Разные общедоступные источники информации неоднократно сообщали, что фирмы США заказывают программное обеспечение (ПО) в Индии на сумму превышающую 17 млрд. долларов в год, в Китае на сумму превышающую 10 млрд. в год, в России 1 млрд. в год. Разработка ПО в этих странах стоит дешевле, чем в США. Тем не менее, в самих США объем разрабатываемого ПО наверняка сравним с приведенными цифрами. Из этой информации можно сделать несколько выводов:

Первый. Парк ПЭВМ в США используется с очень высоким КПД.

Второй. Всякий существенный рост производства провоцирует его усовершенствование с целью обеспечения этого роста, что и привело к организации центров по разработке ПО с целью расширения круга решаемых задач в том числе весьма сложных, посредством создания кластеров при необходимости больших объемов памяти, повышения эффективности работы программистов и конечно же качества ПО.

Третий. Масштаб разработки ПО показывает, что фирмы США в своем большинстве хорошо представляют задачи, которые они могут решать с помощью ПЭВМ и могут сформулировать исходные данные для разработки ПО как самостоятельно, так возможно и с помощью соответствующих центров.

Возможная структурная модель центра по разработке ПО для ПЭВМ должна учитывать технологию разработки ПО и некоторые организационные моменты, способствующие эффективной работе центра:

1. Группа менеджеров, обеспечивающих центр заказами и ведущими финансовыми вопросами.

2. Группа разработчиков алгоритмов, которые должны согласовывать постановку задачи заказчику в виде словесного, словесно-формируемого описания, циклограммы с комментариями и с полным набором исходных данных для разработки алгоритмов и программ, а также разрабатывающих соответствующие алгоритмы и контрольные примеры или мат. модели для тестирования программ, необходимую промежуточную документацию и инструкцию для пользователей.

Алгоритм предназначенный для реализации в ПЭВМ – это логика задействования стандартных процедур с параметрами поставленной задачи в рамках выбранного языка программирования.

Наиболее рациональным является модульный принцип разработки ПО [10], когда задача решается комплексом алгоритмов (если это необходимо) к каждому из которых обозначен перечень исходных данных позволяющий протестировать все без исключения ветви алгоритмов, а в итоге и ПО в целом. Модульный принцип позволяет уменьшить количество ошибок и облегчить отработку ПО в ПЭВМ.

Перечень параметров и их характеристики могут быть представлены в виде таблиц к каждому алгоритму.

Разработка алгоритмов с использованием методики структурного программирования [10, 11] позволяет прозрачно представить выше упомянутую логику, облегчить разработку тестов и инструкцию для пользователей.

3. Группа программистов, которые разрабатывают программы, реализуют их в ПЭВМ, проверяют на правильность функционирования с использованием контрольных примеров или мат. моделей и оформляют документацию по результатам разработки.

4. Группа, которую с некоторой долей условности, можно назвать экспертной, в обязанности которой должно входить – выбор языка программирования, анализ алгоритмов и программ с точки зрения оптимальности разработок как-то объема памяти, времени работы, достаточности тестирования, а также мониторинг публикаций появляющихся в спец. литературе с целью возможного их использования и накопления соответствующей литературы.

5. Архив. Процесс документирования должен сопровождать все этапы разработки и реализации программ. Согласно разным источникам, отражающим практику разработки ПО, здесь как нигде более важно накопление опыта в силу специфики, заключающейся в беспредельном разнообразии и количестве возможных задач из разных областей знания и функционирования общества целом, что как раз и предопределяет схожесть решений во многих случаях. Главное условие документирования – дать возможность людям понять программы, которые написаны другими с учетом еще и того факта, что разработанное ПО может эксплуатироваться многие годы и может возникать необходимость его коррекции или усовершенствования. Задача документирования весьма сложная и качество ее выполнения полностью зависит от квалификации разработчиков ПО. Кроме всего

прочего, согласно принятому Верховной Радой закону, разработанное ПО должно быть интеллектуальной собственностью учредителя соответствующего центра.

Организация центров разработки ПО не требует больших материальных средств (во всяком случае заведомо несравнимых с созданием производств законченного технологического цикла, за исключением разве что отверточных технологий – то бишь сборочных, которые в интеллектуальном смысле любой нации ничего не дают), но зато требует большого интеллектуального багажа и поэтому в США они создавались на базе университетов.

Областей применения ПЭВМ очень много и они общеизвестны. Но здесь стоит обозначить хотя бы некоторые, чтобы показать мощный потенциал этого инструмента в том числе и как средства нового знания.

В первую очередь это конечно же фундаментальная и прикладная наука где множество расчетных задач, моделирование, разработка чертежей, электросхем и т.д. Сюда же можно отнести стыковку ПЭВМ с измерительной техникой такой как контрольная или диагностическая в результате чего получают уникальные результаты, как следствие оперативной обработки результатов измерений (тот же медицинский томограф). Экзотическим примером может служить стыковка ПЭВМ с врачебным стетоскопом в результате чего получают изображение легких человека на экране монитора, что помогает точно определить место поражения легких туберкулезом (Израиль, ФРГ).

Образовательные программы. Значение которых трудно переоценить.

Разработка баз данных. Касается практически всех областей функционирования общества целом и наиболее простая область с точки зрения разработки программ.

В искусстве это кино и даже музыка.

Игровые программы. Наиболее прибыльная область использования ПЭВМ.

ПЭВМ соединяет в себе качества игрушки (без кавычек) и полезного инструмента – первое мало что дает нации в целом, недоиспользование второго означает существенно переплачивать производителю за каждую ПЭВМ и кстати государству в котором этот производитель расположен, поскольку он платит там налоги, уровень которых зависит от масштаба производства.

Литература

1. Смирнов Н.В., Дунин - Барковский И.В. Курс теории вероятностей и математической статистики. М., “Наука”, 1969.
2. Рабинович С.Г. Погрешности измерений Л., “Энергия” 1978.
3. Электрические измерения неэлектрических величин. Под ред. проф. П.В. Новицкого Л., “Энергия”, 1975.
4. Зайдель А.Н. Элементарные оценки ошибок измерений. М., “Наука” 1967.
5. Сквайрс Дж. Практическая физика. М., “Мир” 1971.
6. Новицкий П.В. Основы информационной теории измерительных устройств. Л., “Энергия” 1968.
7. Кавалеров Г.И., Мандельштам С.М. Введение в информационную теорию измерений. “Энергия” 1974.
8. Эльясберг П.Е. Измерительная информация: Сколько ее нужно? Как ее обрабатывать? М., “Наука”, 1983.
9. Любимов А.И., Форсилова И.Д., Шапиро Е.З. Проверка средств электрических измерений. Л., “Энергия”, 1987.
10. Гудман С., Хидетниemi С. Введение в разработку и анализ алгоритмов. М., “Мир”, 1981.
11. Дал У., Дейкстра Э., Хоорк К. Структурное программирование. М., “Мир”, 1975.