

УДК 621.9.025 (076)
ББК 34.63 я7
А 22

Рецензент канд. техн. наук, доцент кафедры «Технология машиностроения» О. Г. Крупенников.

Одобрено секцией методических пособий научно-методического совета университета.

А 22 Автоматизированное проектирование инструментов и инструментальной оснастки : методические указания к контрольным работам / сост. В. В. Демидов. – Ульяновск : УлГТУ, 2015. – 40 с.

Составлены в соответствии с рабочей программой по дисциплине «Автоматизированное проектирование инструментов и инструментальной оснастки» для студентов всех форм обучения направления 15.03.05 «Конструкторско-технологическое обеспечение машиностроительных производств».

В указаниях изложены общие положения по алгоритмизации расчетов режущих инструментов и инструментальной оснастки, формам представления алгоритмов и правилам их выполнения в виде блок-схемы (с примером), методика алгоритмирования и составления сегментов программ для ЭВМ (с примером) при решении задачи определения корней трансцендентного инволютного уравнения. Приведены требования к содержанию и оформлению контрольной работы.

Методические указания подготовлены на кафедре «Металлорежущие станки и инструменты».

УДК 621.9.025(076)
ББК 34.63 я7

© Демидов В.В., составление, 2015
© Оформление. УлГТУ, 2015

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1. ЭТАПЫ ПОДГОТОВКИ И РЕШЕНИЯ ЗАДАЧ НА ЭВМ.....	6
2. НЕКОТОРЫЕ ПРАВИЛА ВЫПОЛНЕНИЯ БЛОК-СХЕМ АЛГОРИТМОВ И СИМВОЛОВ, ИСПОЛЬЗУЕМЫХ В НИХ.....	9
2.1. Правила выполнения блок-схем алгоритмов.....	9
2.2. Правила применения символов в блок-схемах алгоритмов.....	12
2.3. Символы в блок-схемах алгоритмов.....	16
2.4. Соотношение геометрических элементов символов.....	18
3. ЯЗЫКИ ПРОГРАММИРОВАНИЯ.....	20
4. АЛГОРИТМИЗАЦИЯ РЕШЕНИЯ ИНВОЛЮТНОГО УРАВНЕНИЯ МЕТОДОМ ИТЕРАЦИИ.....	22
4.1. Постановка задачи и методы определения корня инволютного уравнения	22
4.2. Определение корня инволютного уравнения шаговым методом изменения значений аргумента	24
4.3. Определение корня инволютного уравнения методом касательных.....	28
4.4. Определение корня инволютного уравнения методом хорд	29
4.5. Определение корня инволютного уравнения методом среднего	30
5. СОДЕРЖАНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ ПО КОНТРОЛЬНОЙ РАБОТЕ.....	31
ПРИЛОЖЕНИЯ А-Г.....	32
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	40

ВВЕДЕНИЕ

В настоящее время одним из основных направлений научно-технического прогресса является автоматизация проектирования. Только с использованием систем автоматизированного проектирования можно обеспечить выполнение возрастающего объема проектно-конструкторских работ при ограниченных людских и материальных ресурсах в приемлемые сроки. При неавтоматизированном проектировании используются методы, основанные на инженерном опыте и интуиции. С ростом сложности проектируемых объектов значительно возрастают сроки и стоимость проектирования, поэтому появилась необходимость привлечения вычислительной техники для решения инженерных задач. Однако первоначальное применение ЭВМ в этих целях требовало от пользователя непосредственного участия в разработке трудоемких этапов математической постановки задачи, выбора метода ее решения, алгоритмизации, программирования и отладки. Существующие в настоящее время системы автоматизированного проектирования позволяют во многих случаях выполнить почти все эти этапы на ЭВМ с помощью заранее подготовленного программного обеспечения, рассчитанного на многократное применение.

Цель контрольной работы – научиться методике алгоритмизации и составления сегментов программ для ЭВМ на примере решения задачи нахождения корня инволютного уравнения.

Контрольная работа выполняется каждым студентом самостоятельно по индивидуальному заданию (приложение А) и оформляется в виде пояснительной записки на листах формата А4. Содержание пояснительной записки указано в разделе 5 методических указаний.

На титульном листе пояснительной записки должно быть указано:

Контрольная работа

по дисциплине «Автоматизированное проектирование инструментов и инструментальной оснастки»

Выполнил студентфакультета, группы, ФИО.....

Проверил преподаватель.....ФИО

Дата _____

Перед выполнением контрольной работы студент должен внимательно изучить данные методические указания, уяснить цель работы и получить письменное задание на контрольную работу от преподавателя.

После выполнения контрольной работы и ее оформления в виде пояснительной записки студент должен защитить работу, отвечая преподавателю на контрольные вопросы и показывая усвоение изученного материала.

1. ЭТАПЫ ПОДГОТОВКИ И РЕШЕНИЯ ЗАДАЧ НА ЭВМ

Решение любой задачи на ЭВМ, в том числе и задачи расчета и конструирования режущих инструментов, это сложный процесс, включающий *несколько этапов* [1]:

- а) постановка задачи (например, рассчитать шевер);
- б) математическая формулировка задачи [описать исходные данные в математической форме; записать условия задачи в виде математических обозначений и формул с различными возможными ограничениями (например, формулы, используемые для расчета шевера)];
- в) выбор метода решения задачи: в большинстве случаев одна и та же задача может решаться несколькими методами (например, шевер может быть рассчитан упрощенным методом как плоская задача с использованием эквивалентного зубчатого колеса и шевера или как пространственная задача зацепления колеса и шевера с винтовыми зубьями при перекрещивающихся осях);
- г) разработка алгоритма решения задачи, или алгоритмизация задачи – первый этап программирования;
- д) написание программы – запись алгоритма на языке программирования; программа – это последовательность команд компьютера, приводящая к решению задачи;
- е) отладка программы – необходима для обнаружения и исправления ошибок в программе, которые могут быть синтаксическими, ошибками выполнения (деление на ноль или извлечение квадратного корня из отрицательного числа), логическими или алгоритмическими. Синтаксические ошибки выявляются при компиляции программы – это простые с точки зрения исправления ошибки. Ошибки выполнения и алгоритмические ошибки программы выявить гораздо труднее. Правильность работы программы проверяется путем ее тестирования – сравне-

ния результатов расчетов по разработанной программе и расчетов, выполненных вручную, или на ЭВМ по другой верифицированной программе;

з) обработка и анализ результатов расчетов по программе ведется, как правило, лицом, в интересах которого решалась задача.

Одним из вышеназванных этапов подготовки и решения задач является разработка алгоритма. Разработка алгоритма, как правило, выполняется специалистом, в интересах которого разрабатывается программа для решения конкретных задач. В частности, разработка алгоритма проектирования металлорежущего инструмента выполняется инженером-конструктором, проектирующим инструмент.

Алгоритм – это последовательность строго сформулированных действий, которая позволяет на основе исходных данных с использованием математической формулировки задачи в соответствии с выбранным методом ее решения привести к однозначному результату. Алгоритм должен обладать четырьмя свойствами:

1. Дискретность (раздельность) – решение задачи осуществляется путем последовательного исполнения простых действий.
2. Определенность (детерминированность) – многократное выполнение алгоритма при одних и тех же исходных данных должно приводить к одному и тому же результату.
3. Результативность (конечность) – возможность получения определенного результата для допускаемых значений исходных данных за конечное число шагов.
4. Универсальность (массовость) – алгоритм должен служить для решения не одной, а целого класса аналогичных задач, различающихся лишь исходными данными.

Если для заданной задачи невозможно составить алгоритм, обладающий указанными свойствами, то задача является алгоритмически неразрешимой и не может быть решена на ЭВМ.

Существуют три формы представления (описания) алгоритмов:

1. Словесная – описание алгоритма на естественном разговорном языке (например, см. описание алгоритма поиска корня α инволютного уравнения с требуемой точностью при заданном значении $\ln v\alpha$ в п. 4.2).
2. Операторная – подробное описание алгоритма с использованием формул, логических и математических выражений (например, см. описание алгоритма поиска корня α инволютного уравнения с требуемой точностью при заданном значении $\ln v\alpha$ в п. 4.2).
3. Графическая – представление алгоритма в виде блок-схемы, в которой совокупность и последовательность действий изображены графически символами (фигурами), соединенными между собой прямыми линиями (например, см. приложение Б). Это наиболее распространенная форма описания алгоритмов.

Практически любой сложный алгоритм представляет собой комбинацию трех типов структур: линейной, разветвленной, циклической.

Линейный алгоритм – последовательность операций (действий) выполняемых только один раз в порядке их следования (рис. 1.1, а).

Разветвленный алгоритм – содержит условие, при выполнении которого необходимо выполнить определенные операции (действия), а при невыполнении – другие операции (действия) (рис. 1.1, б).

Циклический алгоритм – последовательность операций (действий), выполняемых многократно (рис. 1.1, в).

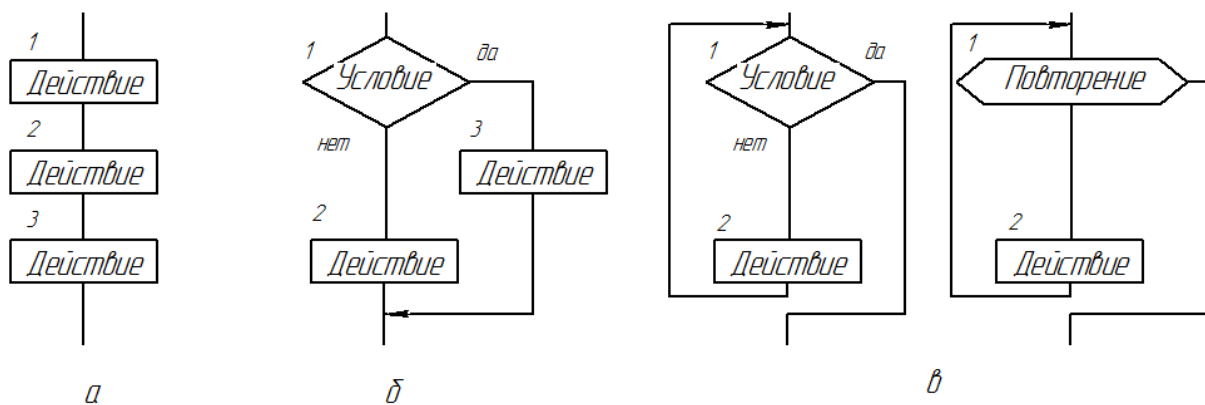


Рис. 1.1. Три типа структур алгоритма: а – линейная; б – разветвленная, в – циклическая

2. НЕКОТОРЫЕ ПРАВИЛА ВЫПОЛНЕНИЯ БЛОК-СХЕМ АЛГОРИТМОВ И СИМВОЛОВ, ИСПОЛЬЗУЕМЫХ В НИХ

Правила выполнения блок-схем алгоритмов и применения символов в них изложены в ГОСТ 19.002-80 [2], а правила на обозначения условные графические изложены в ГОСТ 19.003-80 [3].

2.1. Правила выполнения блок-схем алгоритмов

2.1.1. При выполнении блок-схем алгоритмов отдельные функции алгоритмов с учетом степени их детализации, отображаются в виде условных графических обозначений – символов по ГОСТ 19.003-80 [2]. Схемы должны быть выполнены на форматах по ГОСТ 2.301-68 [4].

2.1.2. Для облегчения вычерчивания и нахождения на схеме символов рекомендуется поле листа разбивать на зоны. Размеры зон устанавливаются с учетом минимальных размеров символов, изображенных на данном листе. Допускается один символ размещать в двух и более зонах, если размер символа превышает размер зоны.

2.1.3. Координаты зоны проставляют: по горизонтали – арабскими цифрами слева направо в верхней части листа;

по вертикали – прописными буквами латинского алфавита сверху вниз в левой части листа.

2.1.4. Координаты зон в виде сочетания букв и цифр присваивают символам, вписанным в поля этих зон, например А1, А2, А3, В1, В2, В3 и т. д. При выполнении схем от руки, если поле листа не разбито на зоны, символам присваивают порядковые номера.

2.1.5. В пределах одной схемы, при выполнении ее от руки, допускается применять не более двух смежных размеров ряда чисел, кратных 5.

2.1.6. Для ускорения выполнения схем от руки рекомендуется использовать бланки с контуром прямоугольника внутри каждой зоны. Контур не должен воспроизводиться при изготовлении копии.

2.1.7. Расположение символов на схеме должно соответствовать требованиям ГОСТ 19.003-80 [3]. Исключение составляют обязательные символы «Линия потока», «Канал связи», «Комментарий» и рекомендуемые символы «Межстраничный соединитель», «Транспортирование носителей», «Материальный поток».

2.1.8. Линии потока должны быть параллельны линиям внешней рамки схемы.

2.1.9. Направления линии потока сверху вниз и слева направо принимают за основные и, если линии потока не имеют изломов, стрелками можно не обозначать. В остальных случаях направление линии потока обозначать стрелкой обязательно.

2.1.10. Расстояние между параллельными линиями потока должно быть не менее 3 мм, между остальными символами схемы – не менее 5 мм.

2.1.11. Записи внутри символа или рядом с ним должны выполняться машинописью с одним интервалом или чертежным шрифтом по ГОСТ 2.304-81 [5].

2.1.12. Записи внутри символа или рядом с ним должны быть краткими.

Сокращения слов и аббревиатуры, за исключением установленных государственных стандартами [6], должны быть расшифрованы в нижней части поля схемы или в документе, к которому эта схема относится.

2.1.13. Для удобства детализации программы должны быть использованы символы «Процесс», «Решение», «Модификация», «Ввод-вывод» и «Пуск-останов», при этом внутри символов на расстояние не менее 0,25 а проводят тонкую линию (размер а по ГОСТ 19.003-80 [3]).

2.1.14. Записи внутри символа должны быть представлены так, чтобы их можно было читать слева направо и сверху вниз, независимо от направления потока (рис. 2.1): вид а должен быть прочитан как вид б.

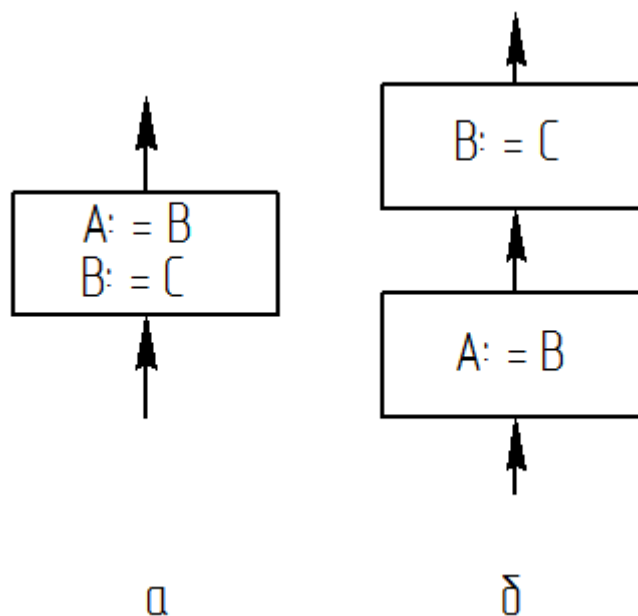


Рис. 2.1. Правило прочтения записей внутри символа

2.1.15. В схеме символу может быть присвоен идентификатор, который должен помещаться слева над символом (например, для ссылки в других частях документации) (рис. 2.2).

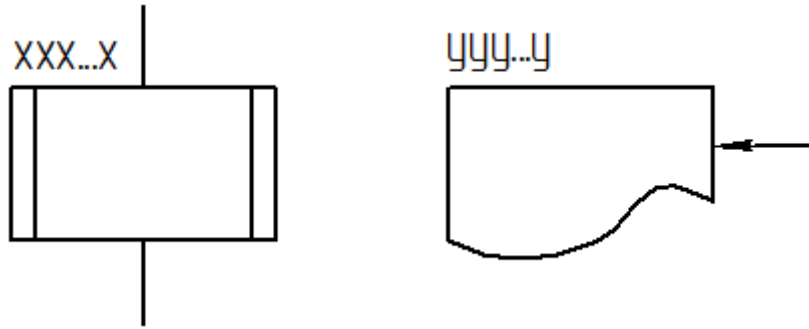


Рис. 2.2. Местоположение идентификатора символа

2.1.16. В схемах допускается краткая информация о символе (описание, уточнение или другие перекрестные ссылки для более полного понимания функции данной части схемы). Описание, уточнение или другие перекрестные ссылки символа должны помещаться справа над символом (рис. 2.3).

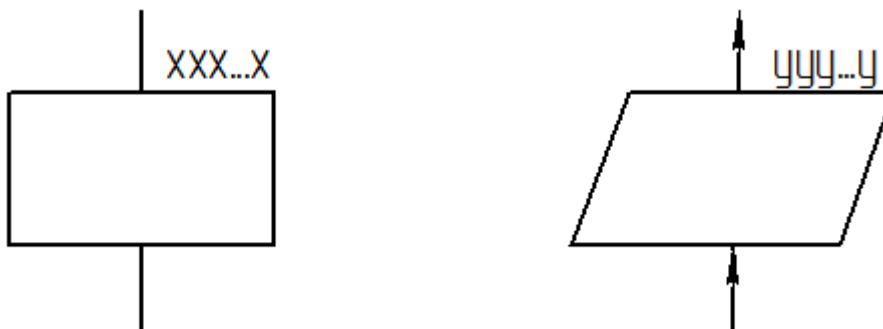
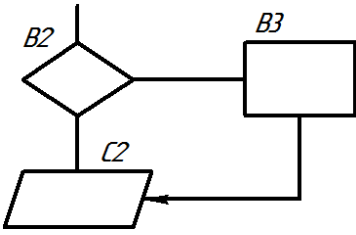
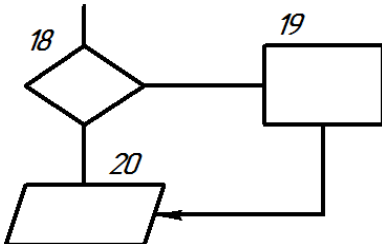
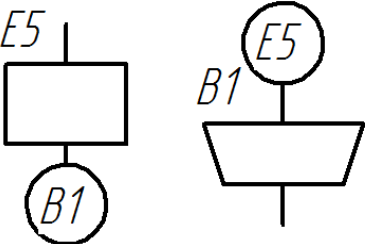
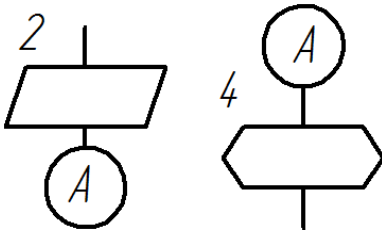
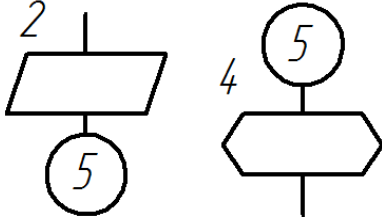


Рис. 2.3. Местоположение краткой информации о символе

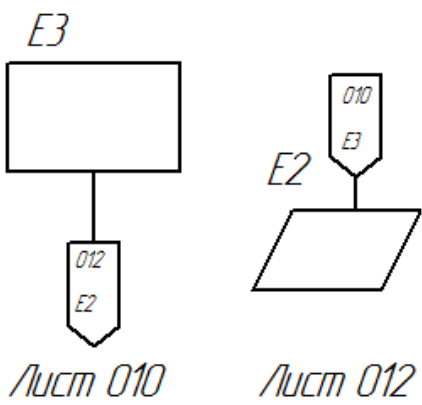
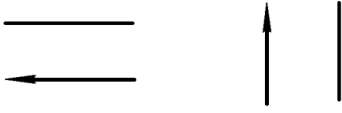
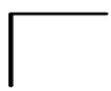
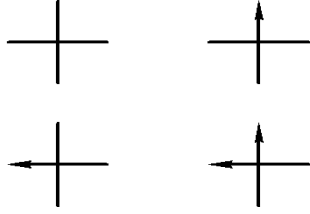
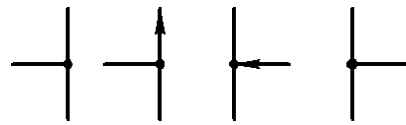
2.2. Правила применения символов в блок-схемах алгоритмов

2.2.1. Применение символов в блок-схемах алгоритмов должно соответствовать указанному в табл. 2.1 [2].

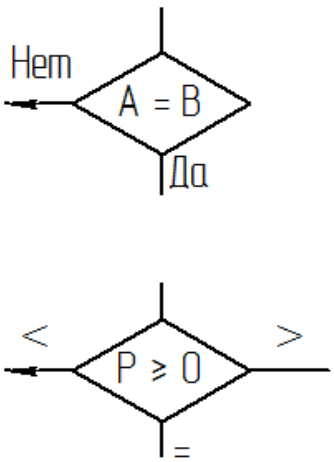
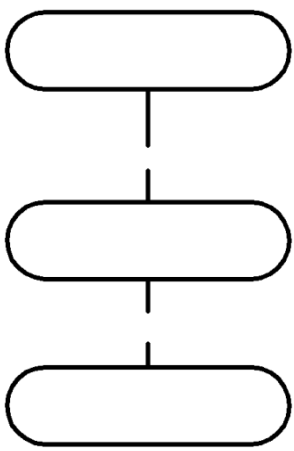
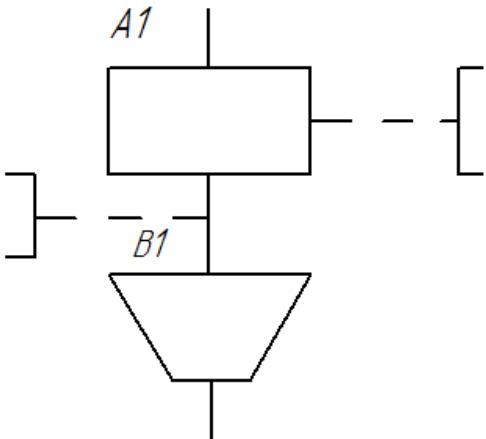
Таблица 2.1. Правила применение символов в блок-схемах алгоритмов

Фрагмент схемы	Содержание обозначения	Правила применения
1	2	3
	<p>Возможные варианты обозначения символов в схемах:</p> <p>B2, B3, C2 – координаты зоны листа, в которой размещен символ;</p>	<p>Координату зоны символа или порядковый номер проставляют слева в верхней части символа в разрыве его контура.</p>
	<p>18, 19, 20 – порядковые номера символов на схеме</p>	
	<p>Соединитель: E5, B1 – идентификаторы соединителя в виде буквы и цифры (координаты зоны листа)</p>	<p>При большой насыщенности схемы символами отдельные линии потока между удаленными друг от друга символами допускается обрывать. При этом в конце и начале обрыва должен быть помещен символ «Соединитель».</p>
	<p>Соединитель: A – идентификатор соединителя в виде буквы</p>	
	<p>Соединитель: 5 – идентификатор соединителя в виде цифры</p>	

Продолжение табл. 2.1

1	2	3
	<p>Межстраничный соединитель.</p> <p>Первая строка внутри межстраничного соединителя определяет номер листа схемы, вторая – координату символа</p>	<p>Связываемые линии потока символы находятся на разных листах.</p> <p>При изготовлении схем с помощью ЭВМ допускается указывать рядом с обрывом линии потока адресные ссылки без использования символов «Соединитель» и «Межстраничный соединитель».</p>
	<p>Линии потока</p>	<p>Применяют для указания направления линии потока: можно без стрелки, если линия направлена слева направо и сверху вниз; со стрелкой – в остальных случаях.</p>
	<p>Излом линии потока под углом 90°</p>	<p>Обозначают изменение направлений линии потока.</p>
	<p>Пересечение линий потока</p>	<p>Применяется в случае пересечения двух несвязанных линий потока.</p>
	<p>Слияние линий потока: место слияния линий потока обозначено точкой</p>	<p>Применяется в случае слияния линий потока, каждая из которых направлена к одному и тому же символу на схеме.</p>

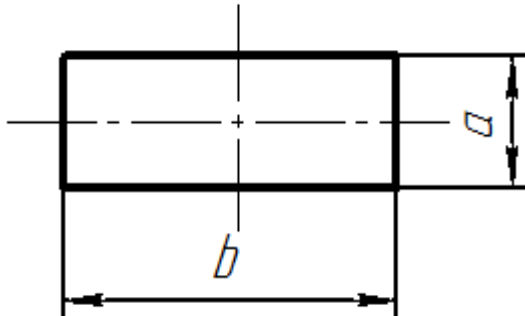
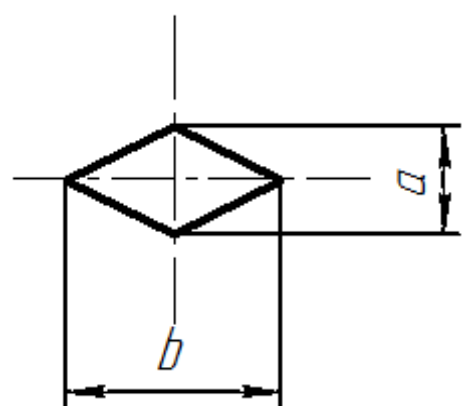
Окончание табл. 2.1

1	2	3
	<p>Возможные варианты отображения решения</p> <p>$A = B$, $P \geq 0$ – условия решений;</p> <p>A, B, P – параметры</p>	<p>При числе исходов не более трех признак условия решения (Да, Нет, =, >, <) проставляют над каждой выходящей линией потока или справа от линии потока.</p>
	<p>Пуск</p> <p>Прерывание</p> <p>Останов</p>	<p>Символы применяют в начале схемы алгоритма или программы, в случае прерывании ее и в конце. Внутри символа «Пуск – останов» может указываться наименование действия или идентификатор программы.</p>
	<p>Комментарий</p>	<p>Применяется, если пояснение не помещается внутри символа (для пояснения характера параметров, особенностей процесса, линий потока и др.). Комментарий записывают параллельно основной надписи. Комментарий помещают в свободном месте схемы алгоритма на данном листе и соединяют с поясняемым символом.</p>

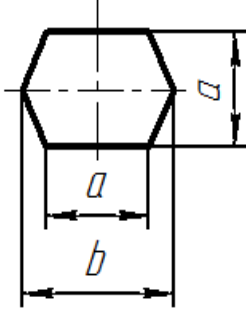
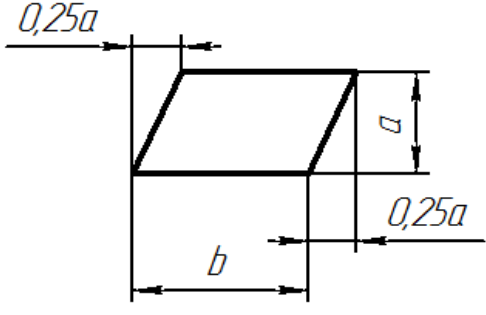
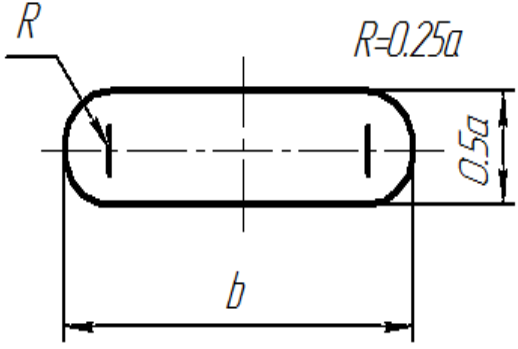
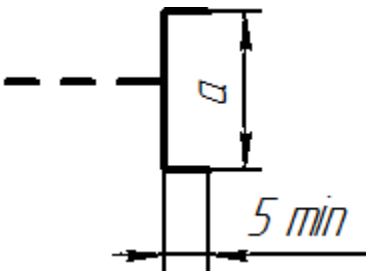
2.3. Символы в блок-схемах алгоритмов

2.3.1. Перечень, наименование, обозначение и размеры символов и отображаемые ими функции в блок-схемах алгоритмов должны соответствовать указанным в табл. 2.2 [3].

Таблица 2.2. Перечень, наименование, обозначение и размеры символов и отображаемые ими функции в блок-схемах алгоритмов

№	Наименование	Обозначение и размеры, мм	Функция
1	2	3	4
1	Процесс		Выполнение операции или группы операций, в результате которых изменяется значение, форма представления или расположение данных.
2	Решение		Выбор направления выполнения алгоритма или программы в зависимости от некоторых переменных условий.

Окончание табл. 2.2

1	2	3	4
3	Модификация		Выполнение операций, меняющих команды или группы команд, изменяющих программу.
4	Ввод – вывод		Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод).
5	Пуск – останов		Начало, конец, прерывание процесса обработки данных или выполнение программы.
6	Комментарий		Связь между элементом схемы и пояснением.

2.4. Соотношение геометрических элементов символов

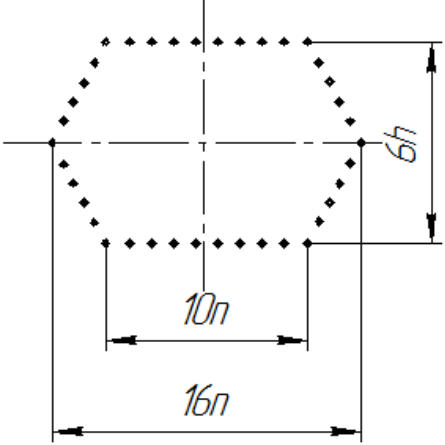
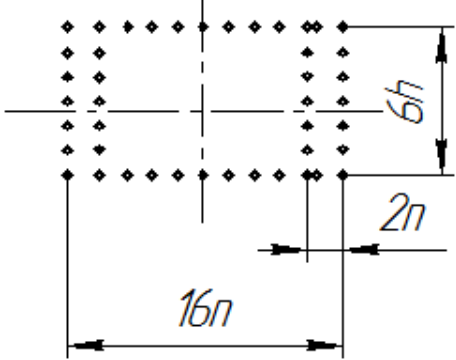
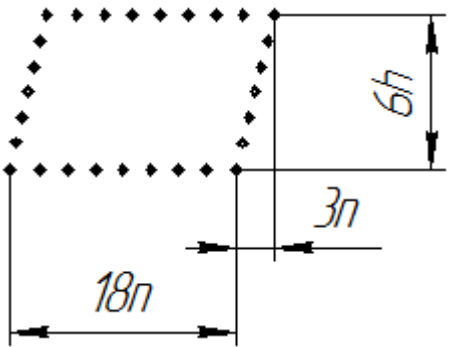
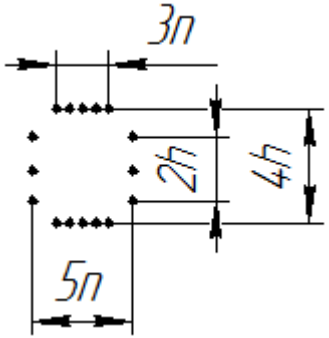
2.4.1. Размер a символов (см. табл. 2.2) должен выбираться из ряда 10, 15, 20 мм. Допускается увеличивать размер a на число, кратное 5. Размер b равен $1,5a$. При ручном выполнении блок-схем алгоритмов для символов 1-3, 4, 5 допускается устанавливать b равным $2a$.

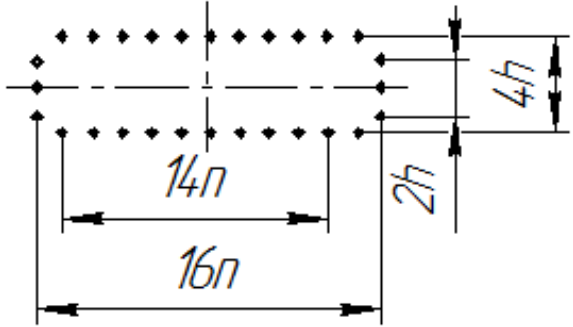
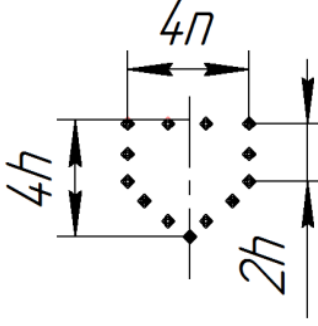

2.4.2. При выполнении условных графических обозначений автоматизированным методом размеры геометрических элементов символов округляются до значений, определяемых техническими возможностями используемых устройств. Символы, выполненные с помощью печатающих устройств, приведены в табл. 2.3, в которой h – шаг печатающего механизма по вертикали; p – шаг печатающего механизма по горизонтали.

Таблица 2.3. Символы, выполняемые с помощью печатающих устройств

№	Наименование	Обозначение
1	2	3
1	Процесс	
2	Решение	

Продолжение табл. 2.3

1	2	3
3	Модификация	 <p>The diagram shows a shape with a horizontal width of $16n$ and a vertical height of $6h$. The top and bottom edges are composed of a series of dots. The top edge is slightly curved, and the bottom edge is also slightly curved. The distance between the two horizontal lines of dots is $10n$.</p>
4	Предопределенный процесс	 <p>The diagram shows a rectangular shape with a horizontal width of $16n$ and a vertical height of $6h$. The shape is filled with a grid of dots. The distance between the two vertical lines of dots is $2n$.</p>
5	Ввод – вывод	 <p>The diagram shows a shape with a horizontal width of $18n$ and a vertical height of $6h$. The shape is filled with a grid of dots. The distance between the two vertical lines of dots is $3n$.</p>
6	Соединитель	 <p>The diagram shows a connector shape with a horizontal width of $5n$ and a vertical height of $4h$. The shape is filled with a grid of dots. The distance between the two vertical lines of dots is $3n$.</p>

1	2	3
7	Пуск – останов	
8	Межстраничный соединитель	
9	Линия потока	

3. ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Язык программирования представляет собой совокупность символов и правил их использования для описания решения задач на ЭВМ [7].

В программировании выделяют языки различных уровней [1] (рис. 3.1): машинные, машинно-ориентированные (ассемблеры), машинно-независимые (языки высокого уровня). Машинные и машинно-ориентированные языки – это языки низкого уровня.

Машинные языки представляют собой систему команд конкретной ЭВМ и реализуется ею непосредственно. Действия, выполняемые машинными ко-

мандами, элементарны: например, переслать содержимое одной ячейки памяти в другую, сложить содержимое двух ячеек и т. д. Запись программы с помощью таких действий требует высокой квалификации и больших трудозатрат. Поэтому на машинном языке программируют редко.

Ассемблеры – промежуточные языки между машинным языком и языком высокого уровня.



Рис. 3.1. Общая классификация языков программирования

Наиболее широко распространено программирование на языках высокого уровня из-за высокой производительности программирования, простоты эксплуатации программ, возможности их переноса с одной машины на другую, наличия средств контроля программ. Специальная программа-транслятор, написанная на внутреннем языке машины, находится в ее памяти и осуществляет перевод программы с языка высокого уровня на внутренний язык машины (машинный).

4. АЛГОРИТМИЗАЦИЯ РЕШЕНИЯ ИНВОЛЮТНОГО УРАВНЕНИЯ МЕТОДОМ ИТЕРАЦИИ

4.1. Постановка задачи и методы определения корня инволютного уравнения

Очень часто при проектировании зуборезных инструментов, применяемых для нарезания зубчатых колес с эвольвентным профилем, приходится определять корень инволютного уравнения [8].

$$\operatorname{inv}\alpha_0 = \operatorname{tg}\alpha - \alpha, \quad (4.1)$$

где α_0 – угол профиля зуба, при котором вычислено значение инволюты $\operatorname{inv}\alpha_0$ (α_0 – точный корень инволютного уравнения); α – искомый угол профиля зуба (в радианах), при котором равенство (4.1) выполняется с требуемой точностью.

Определить корень инволютного уравнения (4.1) – это значит, что при известном значении инволюты $\operatorname{inv}\alpha_0$ необходимо определить (найти) такое значение угла α , при котором равенство (4.1) выполнялось бы с требуемой точностью. Выражение (4.1) представляет собой трансцендентное уравнение, корень которого не может быть представлен в явном виде, а может быть определен лишь приближенно с требуемой точностью. До появления ЭВМ решение таких уравнений было трудоемким и с низкой точностью.

Решение трансцендентных уравнений на ЭВМ осуществляется методом последовательных приближений (методом итерации), который может быть применен для функций, непрерывно и монотонно изменяющихся на выбранном интервале поиска корня [9]. Существуют различные методы приближения к точному корню уравнения α_0 путем изменения значений аргумента [9]: шаговый, касательных, хорд, среднего.

При использовании любого из этих методов необходимо предварительно знать либо приближенное значение корня α уравнения (при шаговом и касательном методах), либо интервал (отрезок) поиска значений аргумента α , в пределах которого находится точный корень α_0 уравнения (для методов хорд и среднего). Приближенное значение корня α уравнения или интервал значений аргумента α , в пределах которого находится точный корень α_0 уравнения, определяется специалистом по рассчитываемому инструменту на основе имеющейся у него информации. Например, из теории зацепления зубчатых колес с эвольвентным профилем известно, что угол α профиля зуба может изменяться от нуля (у основания зуба) до 40° (у вершины зуба), а максимально возможное значение $\alpha = 90^\circ$. Иногда в затруднительных случаях для приближенного определения корня уравнения можно воспользоваться графическим методом (рис. 4.1).

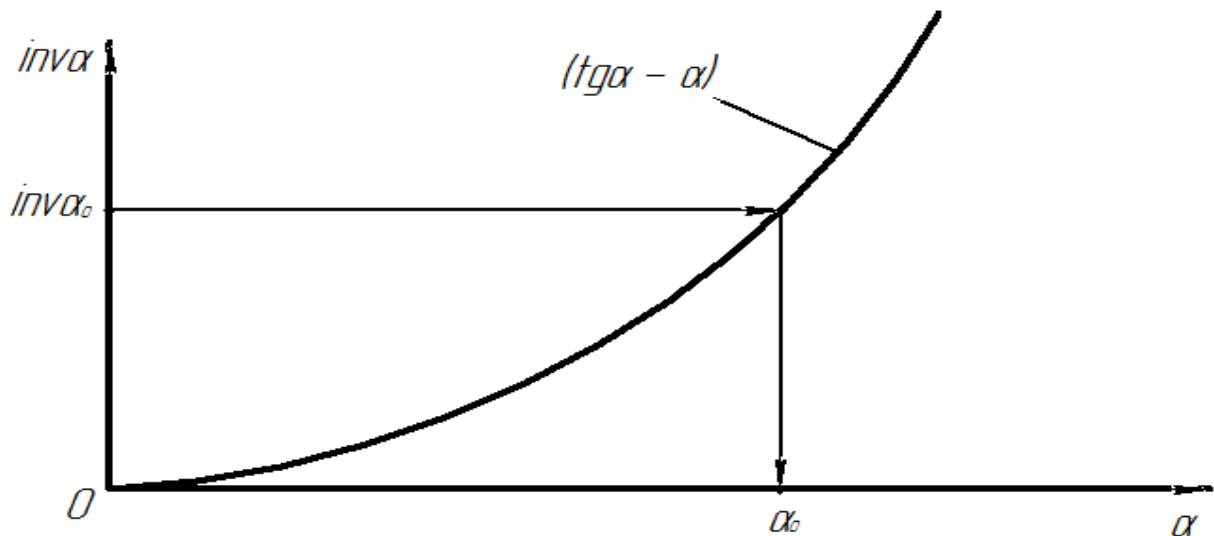


Рис. 4.1. Графическое решение трансцендентного инволютного уравнения (4.1):

α_0 — точный корень инволютного уравнения при известном значении его инволюты $inv\alpha_0$

После определения приближенного значения корня уравнения или интервала поиска значений аргумента приступают к определению корня с требуемой точностью одним из названных выше методов. Организация процесса поиска решения инволютного уравнения с требуемой точностью возможна по двум вариантам:

1-й вариант – либо путем поиска такого значения угла α , при котором абсолютная величина разности значений левой и правой частей уравнения (4.1) стала бы равна с требуемой точностью ε левой части, т. е. $|\operatorname{inv}\alpha_0 - (\operatorname{tg}\alpha - \alpha)| \leq \varepsilon$;

2-й вариант – либо представлением уравнения (4.1) в виде некоторой функции $f(\alpha) = \operatorname{tg}\alpha - \alpha - \operatorname{inv}\alpha_0$, для которой корнем будет такое значение угла α , при котором абсолютное значение функции $f(\alpha)$ будет меньше или равно заданной точности ε нахождения корня, т. е. $|f(\alpha)| \leq \varepsilon$.

Функция $f(\alpha) = \operatorname{tg}\alpha - \alpha - \operatorname{inv}\alpha_0$ является непрерывной и монотонно изменяющейся (возрастающей). Следовательно, для поиска корня инволютного уравнения с требуемой точностью можно воспользоваться одним из вышеназванных методов.

4.2. Определение корня инволютного уравнения шаговым методом изменения значений аргумента

Графическая интерпретация поиска корня инволютного уравнения (4.1) с требуемой точностью ε шаговым методом изменения значений аргумента по 2-му варианту приведена на рис. 4.2, где инволютная функция представлена в виде кривой $f(\alpha) = \operatorname{tg}\alpha - \alpha - \operatorname{inv}\alpha_0$.

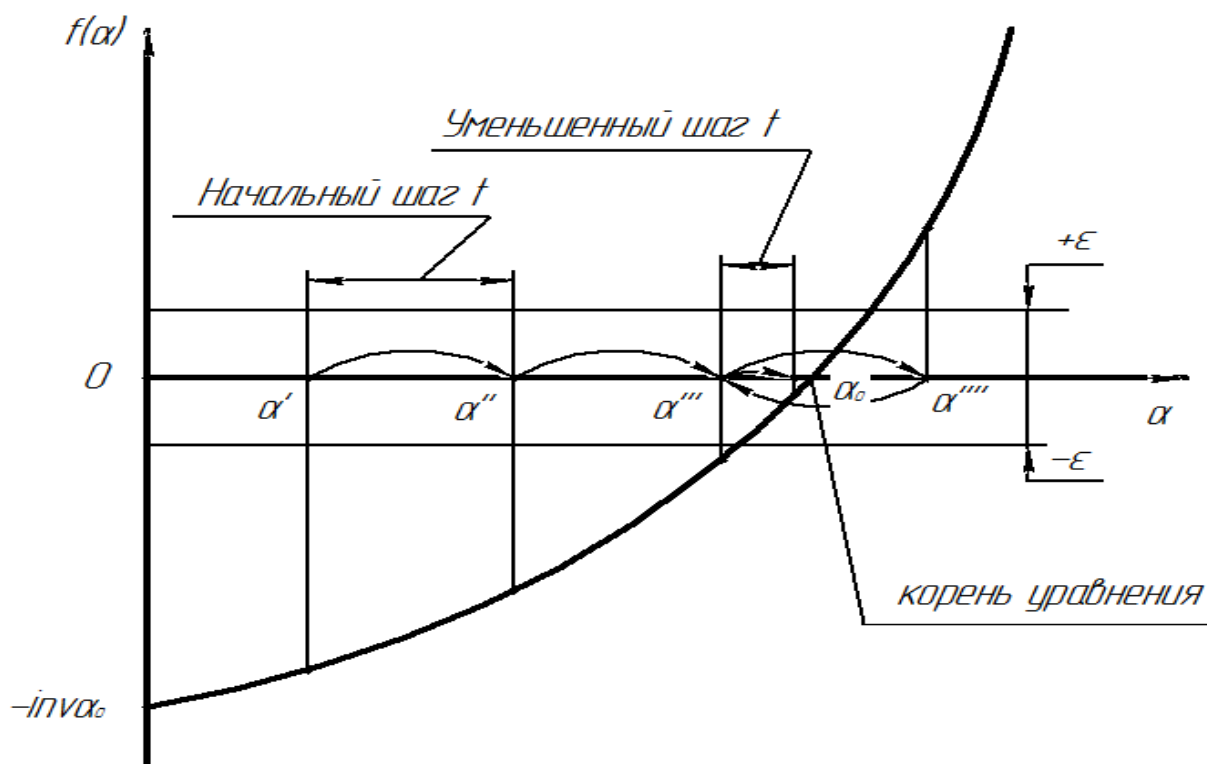


Рис. 4.2. Графическая интерпретация поиска корня инволютного уравнения с требуемой точностью ε шаговым методом изменения значений аргумента по 2-му варианту

Суть метода в следующем. При некотором «грубом» значении корня α' инволютного уравнения (4.1) находим новое более точное его решение α'' по формуле

$$\alpha'' = \alpha' \pm t, \quad (4.2)$$

где t – шаг поиска корня.

Словесное описание алгоритма поиска корня инволютного уравнения (4.1) с требуемой точностью ε шаговым методом по 2-му варианту:

1. Запустить программу поиска корня инволютного уравнения с требуемой точностью шаговым методом.
2. Задаться начальным значением угла α (в градусах), начальным шагом t (в градусах) поиска корня, требуемой точностью ε решения задачи.

3. Преобразовать значения угла α и шага t из градусной меры в радианную.
4. Определить значение функции $f(\alpha)$ при заданном начальном значении угла и проверить условие $|f(\alpha)| \leq \varepsilon$: если условие выполнено, то корень инволютного уравнения с требуемой точностью найден; иначе выполнять действия по п. 5.
5. Определить знак функции $f(\alpha)$ при заданном начальном значении угла α .
6. Увеличивать при отрицательном знаке функции $f(\alpha)$ по п. 5 или уменьшать при положительном знаке функции $f(\alpha)$ по п. 5 значение угла α на шаг t , проверяя для каждого нового значения угла α условие $|f(\alpha)| \leq \varepsilon$: если условие $|f(\alpha)| \leq \varepsilon$ выполнено, то корень инволютного уравнения с требуемой точностью найден; иначе выполнять действия по п. 7.
7. Если при новом значении угла α знак функции $f(\alpha)$ не сменился на противоположный, то повторить действия по п. 6 с шагом t ; если при новом значении угла α знак функции $f(\alpha)$ сменился на противоположный, то целесообразно вернуться к предыдущему значению угла α , уменьшить шаг t , например, в два раза и повторить действия по п. 6 с уменьшенным шагом t .
8. Преобразовать значение найденного корня инволютного уравнения угла α из радианной меры в градусную.
9. Вывести значение найденного корня инволютного уравнения угла α либо на экран, либо на печать, либо в файл.
10. Остановить выполнение программы.

Операторное описание алгоритма поиска корня инволютного уравнения (4.1) с требуемой точностью ε шаговым методом по 2-му варианту (пример):

1. Запустить программу поиска корня инволютного уравнения с требуемой точностью шаговым методом.
2. Присвоить: $\text{inv}\alpha_0=0,014904$, $\alpha=10^\circ$, $t=5^\circ$, $\varepsilon=0,000001$.
3. Вычислить: $\alpha=\alpha\cdot\pi/180$; $\alpha=\alpha\cdot\pi/180$; $t=t\cdot\pi/180$.
4. Вычислить: $f(\alpha)=\text{tg}\alpha-\alpha-\text{inv}\alpha_0$.
5. Если $|f(\alpha)|\leq\varepsilon$, то корень инволютного уравнения с требуемой точностью найден; иначе выполнять действия по п. 6.
6. Если $f(\alpha)<0$, то $\text{znak}=1$, иначе $\text{znak}=-1$.
7. Вычислить: $\alpha=\alpha+\text{znak}\cdot t$.
8. Вычислить: $f(\alpha)=\text{tg}\alpha-\alpha-\text{inv}\alpha_0$.
9. Если $|f(\alpha)|\leq\varepsilon$, то корень инволютного уравнения с требуемой точностью найден; иначе выполнить действия по п. 10.
10. Если $f(\alpha)\cdot f(\alpha-t)<0$, то $\alpha=\alpha-\text{znak}\cdot t$, $t=t/2$, идти к п. 7
11. Вычислить: $\alpha=\alpha\cdot 180/\pi$.
12. Вывести значение найденного корня инволютного уравнения угла α либо на экран, либо на печать, либо в файл.
13. Остановить выполнение программы.

Блок-схемное представление алгоритма выполнить самостоятельно.

4.3. Определение корня инволютного уравнения методом касательных

Графическая интерпретация поиска корня инволютного уравнения с требуемой точностью ε методом касательных по 2-му варианту приведена на рис. 4.3, где инволютная функция представлена в виде кривой $f(\alpha) = \operatorname{tg}\alpha - \alpha - \operatorname{inv}\alpha_0$.

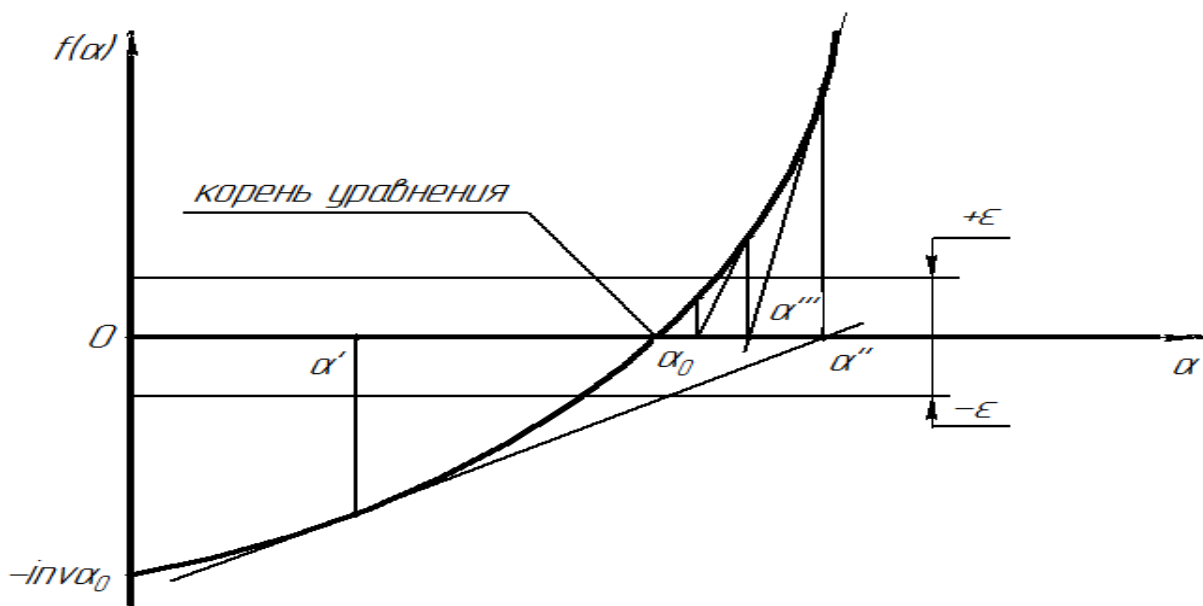


Рис. 4.3. Графическая интерпретация поиска корня инволютного уравнения с требуемой точностью ε методом касательных по 2-му варианту

Суть метода в следующем. При некотором «грубом» значении корня α' инволютного уравнения (4.1) находим новое более точное его решение α'' по формуле

$$\alpha'' = \alpha' - \frac{f(\alpha')}{f'(\alpha')}, \quad (4.3)$$

где $f'(\alpha')$ – значение производной функции $f(\alpha)$ при $\alpha = \alpha'$:

$$f'(\alpha') = \frac{1}{\cos^2(\alpha')} - 1. \quad (4.4)$$

Если при новом значении корня α'' условие $|f(\alpha'')| \leq \varepsilon$ не выполняется, то заменой в формулах (4.3 и 4.4) α' на α'' может быть получено новое более точное решение α''' уравнения (4.1) и т. д. до достижения требуемой точности ε .

4.4. Определение корня инволютного уравнения методом хорд

Графическая интерпретация поиска корня инволютного уравнения с требуемой точностью ε методом хорд по 2-му варианту приведена на рис. 4.4, где инволютная функция представлена в виде кривой $f(\alpha) = \operatorname{tg}\alpha - \alpha - \operatorname{inv}\alpha_0$.

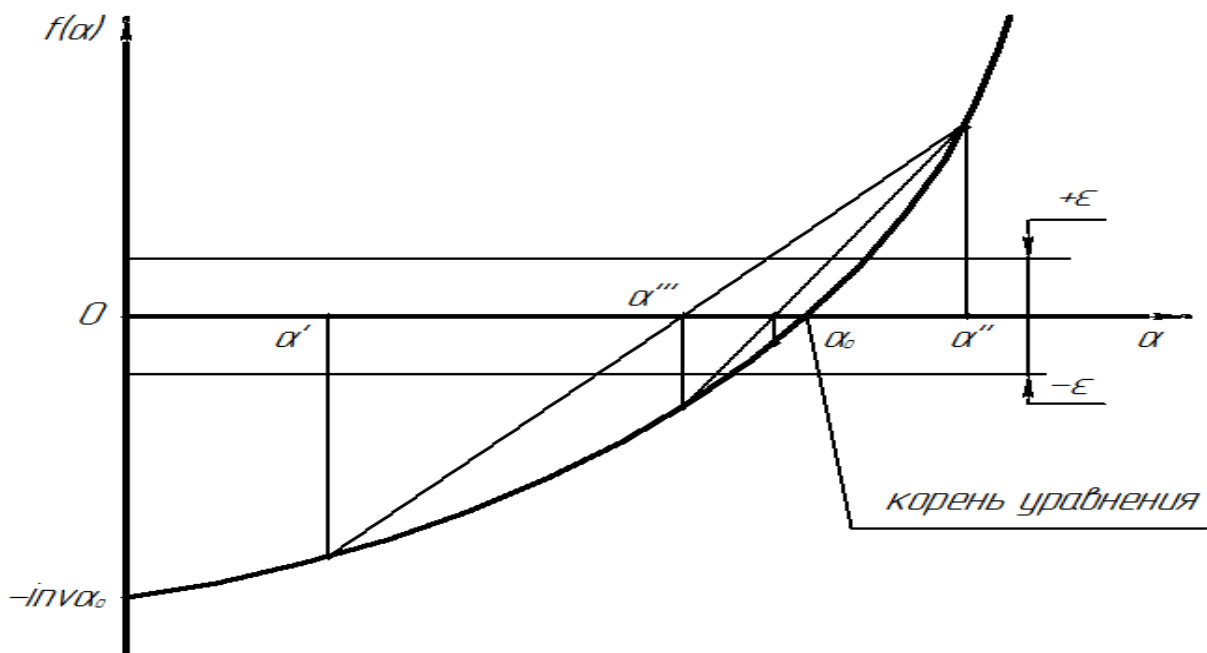


Рис. 4.4. Графическая интерпретация поиска корня инволютного уравнения с требуемой точностью ε методом хорд по 2-му варианту

При известных крайних значениях интервала α' и α'' , на котором может быть корень уравнения (4.1), находим новое его решение α''' по формуле

$$\alpha''' = \alpha' - f(\alpha') \cdot \frac{\alpha'' - \alpha'}{f(\alpha'') - f(\alpha')} \quad (4.5)$$

Если при новом значении корня α''' условие $|f(\alpha''')| \leq \varepsilon$ не выполняется, то заменой в формуле (4.5) α' на α''' (см. рис. 4.4) может быть получено новое более точное решение уравнения (4.1) и т. д. до достижения требуемой точности ε .

4.5. Определение корня инволютного уравнения методом среднего

Поиск корня инволютного уравнения с требуемой точностью ε методом среднего подобен вышерассмотренному поиску корня уравнения методом хорд (см. п. 4.3), в котором новое значение корня уравнения (4.1) α''' находится как среднее арифметическое углов α' и α'' (рис. 4.5).

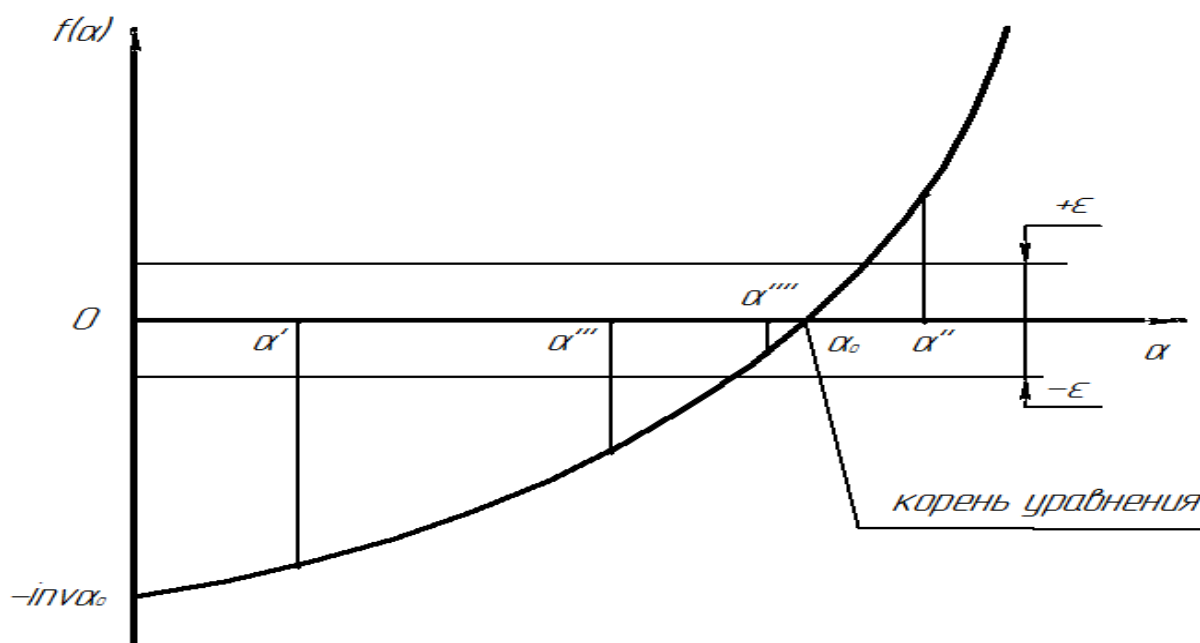


Рис. 4.5. Графическая интерпретация поиска корня инволютного уравнения с требуемой точностью ε методом среднего по 2-му варианту

5. СОДЕРЖАНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ ПО КОНТРОЛЬНОЙ РАБОТЕ

5.1. Задание преподавателя [например, найти корень инволютного уравнения (4.1) с требуемой точностью ε шаговым методом по 1-му или 2-му варианту].

5.2. Графическая интерпретация поиска корня инволютного уравнения (4.1) с требуемой точностью ε указанным в п. 5.1 методом.

5.3. Блок-схема алгоритма определения корня инволютного уравнения (4.1) с требуемой точностью ε указанным в п. 5.1 методом.

5.4. Текст программы на языке Бейсик (Basic) или на другом алгоритмическом языке определения корня инволютного уравнения (4.1) с требуемой точностью ε , разработанной на основе блок-схемы алгоритма по п. 5.3.

5.5. Результаты контрольного поиска корня инволютного уравнения с требуемой точностью указанным методом по разработанной программе (контрольный счет выполняется в присутствии преподавателя по заданному им значению инволюты).

В приложении В приведены основные символы и операторы языка программирования Бейсик, а также порядок выполнения арифметических выражений в программах на этом языке.

В приложении Г приведены примеры текста начала и конца программы определения корня инволютного уравнения методом среднего, результата определения корня инволютного уравнения по приведенной программе и таблицы соответствия обозначений параметров в тексте записки их идентификаторам в блок-схеме алгоритма и программе.

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»**

Кафедра «Металлорежущие станки и инструменты»

ЗАДАНИЕ

на контрольную работу по дисциплине «АПИ и ИО»

студенту машиностроительного факультета группы _____

_____ (Ф.И.О.)

Найти корень инволютного уравнения с требуемой точностью

методом _____ по _____ варианту.

Для этого выполнить следующее:

1. Изобразить графическую интерпретацию поиска корня инволютного уравнения с требуемой точностью указанным методом.
2. Разработать блок-схему алгоритма определения корня инволютного уравнения с требуемой точностью указанным методом.
3. На основании блок-схемы алгоритма по п. 2 разработать программу определения корня инволютного уравнения с требуемой точностью указанным методом на языке Basic (или на другом алгоритмическом языке).
4. Контрольный поиск корня инволютного уравнения с требуемой точностью указанным методом по разработанной программе (контрольный счет выполняется в присутствии преподавателя по заданному им значению инволюты).

Преподаватель _____ (Ф.И.О.)

Студент _____ (Ф.И.О.)

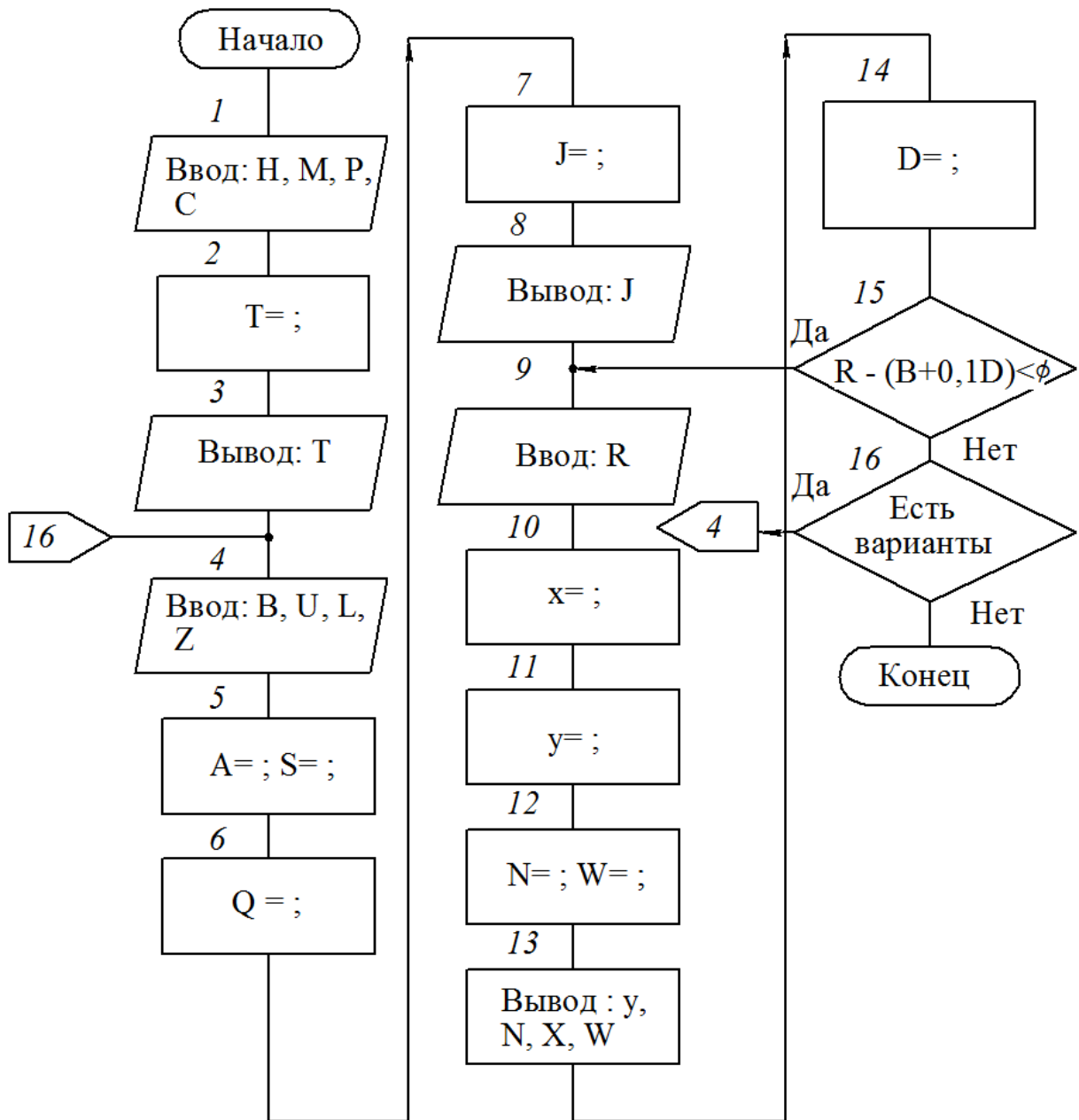


Рис. П.Б1. Пример выполнения некоторого алгоритма в виде блок-схемы

ОСНОВНЫЕ СИМВОЛЫ ЯЗЫКА ПРОГРАММИРОВАНИЯ БЕЙСИК (BASIC)

Алфавит языка Бейсик состоит из 26 латинских прописных букв A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z.

В Бейсике также используются:

- цифры: Ø (ноль), 1, 2, 3, 4, 5, 6, 7, 8, 9 (ноль в тексте программы перечеркивается для его отличия от буквы O);
- знаки: ”–“, “+”, “–“, “*”, ”/”, ”–”, ”=”, “>”, “>=”, “<”, “<=”, “<>”;
- ограничители: “ . “ – десятичная точка; “ , “ – запятая; “ ; “ – точка с запятой; “ : “ – двоеточие; “ ” ” – кавычки; “ (“ – левая скобка; “) ” – правая скобка; “ _ ” – пробел (пустое место);
- константы могут быть представлены в естественной и полулогарифмической формах (рис. П.В1):

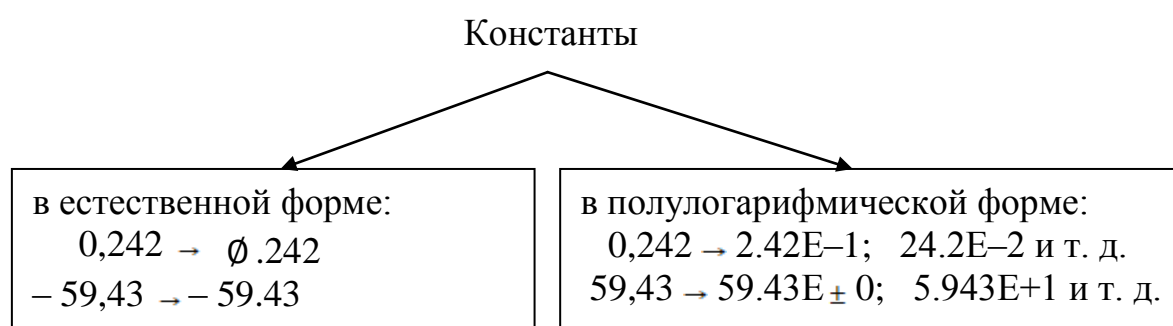


Рис. П.В1. Формы представления констант

- переменные обозначают именем или идентификатором, который состоит из буквы (например, A, B, C и т. д.) или буквы с цифрой (например, A1, A2, B9, D3, и т. д.);

- переменные обозначают именем или идентификатором, который состоит из буквы (например, А, В, С и т. д.) или буквы с цифрой (например, А1, А2, В9, D3, и т. д.);

- стандартные функции:

SIN(X); COS(X); ATN(X), где угол X в радианах;

EXP(X) → e^x ; LOG(X) → $\ln x$; SQR(X) → \sqrt{X} ; ABS(X) → $|X|$;

INT(X) – целая часть числа, не больше самого числа:

например, INT(5.2) = 5; INT(-4.9) = -4; SGN(X) – знак числа при $X < 0 \rightarrow -1$, при $X = 0 \rightarrow 0$, при $X > 0 \rightarrow +1$.

ОСНОВНЫЕ ОПЕРАТОРЫ ЯЗЫКА ПРОГРАММИРОВАНИЯ БЕЙСИК

- оператор присваивания LET: например, LET X=Y+2 или LET X=X+1;
- оператор остановки STOP – логическое завершение программы;
- оператор конца END – физическое завершение программы;
- оператор ввода INPUT: например, INPUT A, B, K<BK>(возврат каретки);
- оператор печати PRINT <цифры или идентификаторы через точку с запятой>; например, при выполнении фрагмента программы:

10 LET A=3

20 LET B= -10

30 PRINT 1; A; 2; B

– по оператору “30 PRINT 1; A; 2; B” будет выведено в строку:

1 3 -10

– если будет оператор вида “30 PRINT “Результат =” 1; A; 2; B”,

то будет выведено в строку: Результат = 1 3 -10

- оператор комментария для снабжения отдельных частей больших и сложных программ поясняющими текстами: REM

(например, 100 REM Расчет высоты зуба);

- оператор безусловного перехода GOTO (например, переход к оператору с номером 50: 20 GOTO 50);

- условный оператор:

три формы написания,

например:

```
5 IF A < B+5 THEN LET G = G*0.5-1: GOTO 20
```

```
5 IF A < B+5 THEN 50
```

```
5 IF A < B+5 GOTO 50 ;
```

- оператор цикла, например, при выполнении фрагмента программы:

```
10 FOR I=0 TO 9
```

```
20 PRINT "I=" I
```

```
30 NEXT I
```

будет написано I=0

I=1

I=2 и т. д. до I=9;

- операторы входа в подпрограмму GOSUB и выхода из подпрограммы RETURN, например, при выполнении фрагмента программы, расположенного внутри основной программы:

Основная программа



.....

```
80 GOSUB 100 (отправляет к подпрограмме, которая начинается с 100 оператора)
```

```
90 STOP
```

подпрограмма {
100
110
... ..
160 RETURN
(возврат на следующий
оператор после оператора GOSUB)

Команда запуска ЭВМ на выполнение программы: RUN.

ПОРЯДОК ВЫПОЛНЕНИЯ АРИФМЕТИЧЕСКИХ ВЫРАЖЕНИЙ В ПРОГРАММАХ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ БЕЙСИК

1. Выполняются операции внутри скобок, начиная с внутренних.
2. Внутри скобок следующий приоритет операций: вычисление значений функций, возведение в степень, умножение и деление, сложение и вычитание.
3. При наличии нескольких операций одного приоритета вычисления выполняются последовательно слева направо.

Например, выражение $\frac{a \cdot x^2 + b \cdot x + c}{d - 2,5}$ в программе следует написать в виде $(A \cdot X^2 + B \cdot X + C) / (D - 2.5)$.

**ПРИМЕР НАЧАЛА И КОНЦА ПРОГРАММЫ ОПРЕДЕЛЕНИЯ
КОРНЯ ИНВОЛЮТНОГО УРАВНЕНИЯ МЕТОДОМ СРЕДНЕГО
НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ БЕЙСИК**

```

10 PRINT "ПРОГРАММА ОПРЕДЕЛЕНИЯ КОРНЯ ИНВОЛЮТНОГО УРАВНЕНИЯ
МЕТОДОМ СРЕДНЕГО"
20 PRINT "ИНВОЛЮТНОЕ УРАВНЕНИЕ В ВИДЕ: f (альфа)=tg(альфа)-альфа-
inv(альфа)
30 PRINT "ВВЕДИТЕ ИСХОДНЫЕ ДАННЫЕ:"
40 PRINT "ЗНАЧЕНИЕ INVALF"
50 INPUT INVALF
60 PRINT "ТРЕБУЕМАЯ ТОЧНОСТЬ ОПРЕДЕЛЕНИЯ КОРНЯ EPS"
70 INPUT EPS
80 PRINT "ИНТЕРВАЛ ПОИСКА КОРНЯ: ALFMIN, ALFMAX"
90 INPUT ALFMIN, ALFMAX
.....
.....
600 PRINT #1, "ПРИ СЛЕДУЮЩИХ ИСХОДНЫХ ДАННЫХ"
610 PRINT #1, "inv(альфа)="; INVALF; "EPS ="; EPS
620 PRINT #1, "ALFMIN ="; ALFMIN; "ALFMAX ="; ALFMAX
630 PRINT #1, "ПОЛУЧЕНЫ СЛЕДУЮЩИЕ РЕЗУЛЬТАТЫ : "
640 PRINT #1, "КОРЕНЬ УРАВНЕНИЯ ALF (градус) ="; ALF
650 PRINT #1, " "
660 CLOSE #1,
670 END

```

**РЕЗУЛЬТАТ ОПРЕДЕЛЕНИЯ КОРНЯ ИНВОЛЮТНОГО
УРАВНЕНИЯ ПО ВЫШЕПРИВЕДЕННОЙ ПРОГРАММЕ**

ПРИ СЛЕДУЮЩИХ ИСХОДНЫХ ДАННЫХ:
 $\text{Inv(альфа)}=0.014904$; $\text{EPS} =0.0001$
ПОЛУЧЕНЫ СЛЕДУЮЩИЕ РЕЗУЛЬТАТЫ:
КОРЕНЬ УРАВНЕНИЯ ALF (градус) = 20

**СООТВЕТСТВИЕ ОБОЗНАЧЕНИЙ ПАРАМЕТРОВ В ТЕКСТЕ
ЗАПИСКИ ИХ ИДЕНТИФИКАТОРАМ В БЛОК-СХЕМЕ
АЛГОРИТМА И ПРОГРАММЕ**

Таблица П.Г1. Соответствие обозначений параметров в тексте записки их идентификаторам в блок-схеме алгоритма и программе определения корня инволютного уравнения методом среднего

В тексте записки	В блок-схеме алгоритма	В программе
$\text{inv}\alpha_0$	$\text{inv}\alpha_0$	INVALF
ε	ε	EPS
α' , градус	αmindeg	ALFMIN
α'' , градус	αmaxdeg	ALFMAX
□□□	□□□	□□□

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кадырова, Г. Р. Основы алгоритмизации и программирования : учебное пособие / Г. Р. Кадырова. – Ульяновск : УлГТУ, 2014. – 95 с.
2. ГОСТ 19.002-80 Схемы алгоритмов и программ. Правила выполнения. – М. : Изд-во стандартов, 1980. – 10 с.
3. ГОСТ 19.003-80 Схемы алгоритмов и программ. Обозначения условные графические. – М. : Изд-во стандартов, 1980. – 12 с.
4. ГОСТ 2.301-68 ЕСКД. Форматы.– М.: Стандартиформ, 2007.–12 с.
5. ГОСТ 2.304-81 ЕСКД. Шрифты чертежные. – М. : ИПК Изд-во стандартов, 2007. – 21 с.
6. ГОСТ Р 7.0.12–2011 СИБИД. Библиографическая запись. Сокращение слов и словосочетаний на русском языке. Общие требования и правила. – М. : Стандартиформ, 2012. – 24 с.
7. Сафронов, И. К. Бейсик в задачах и примерах / И. К. Сафронов. – 2-е изд. – Санкт-Петербург : БХВ-Петербург, 2008. – 320 с.
8. Расчет и конструирование дисковых прямозубых долбяков : методические указания к практическим занятиям по дисциплине «Режущий инструмент» для студентов всех форм обучения направления 151900.62 «Конструкторско-технологическое обеспечение машиностроительных производств» / сост. : В. В. Демидов, Г. И. Киреев. – Ульяновск : УлГТУ, 2014. – 50 с.
9. Выгодский, М. Я. Справочник по высшей математике / М. Я. Выгодский. – М. : АСТ Астрель, 2006. – 991 с.

Учебное издание

Автоматизированное проектирование инструментов и инструментальной оснастки

Методические указания к контрольным работам

Составитель ДЕМИДОВ Валерий Васильевич

Редактор Н. А. Евдокимова
Подписано в печать 30.03.2015. Формат 60×84/16.
Усл. печ. л. 2,32. Тираж 50 экз. Заказ 351.

Ульяновский государственный технический университет
432027, г. Ульяновск, ул. Сев. Венец, д. 32.
ИПК «Венец» УлГТУ, 432027, г. Ульяновск, ул. Сев. Венец, д. 32.