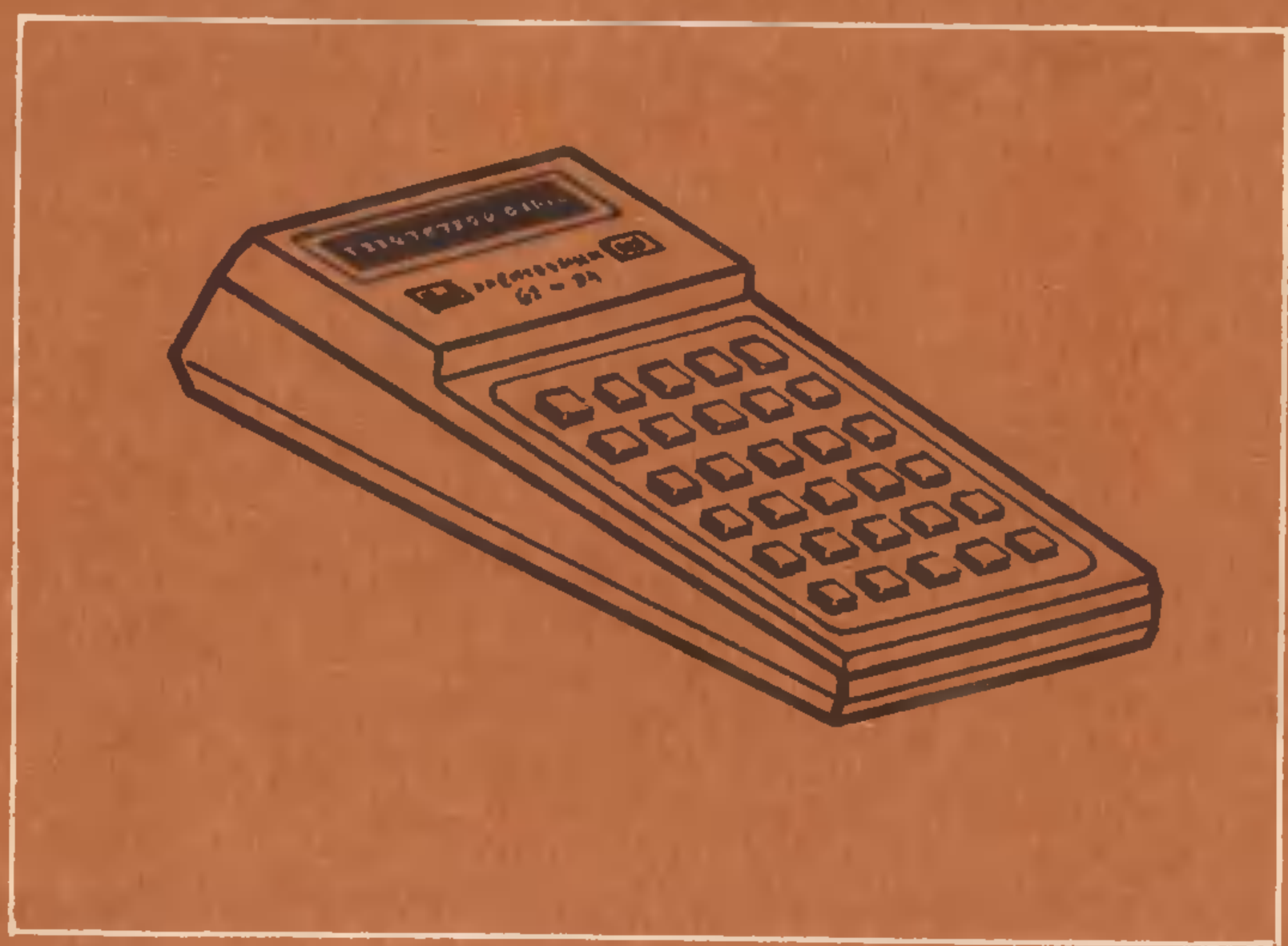
621
С 24

Э. А. Свидерский

РЕШЕНИЕ ТЕХНОЛОГИЧЕСКИХ ЗАДАЧ В МАШИНОСТРОЕНИИ С ПРИМЕНЕНИЕМ МИКРО-КАЛЬКУЛЯТОРОВ



Э. А. Свидерский

РЕШЕНИЕ ТЕХНОЛОГИЧЕСКИХ ЗАДАЧ В МАШИНОСТРОЕНИИ С ПРИМЕНЕНИЕМ МИКРО-КАЛЬКУЛЯТОРОВ

МОСКВА « МАШИНОСТРОЕНИЕ » 1987

681

ББК 34.5:32.973
С24
УДК 621.9.001:681.322

Рецензент д-р техн. наук проф. *Б.И. Белов*

Свидерский Э.А.
С24 Решение технологических задач в машиностроении с применением микрокалькуляторов. – М.: Машиностроение, 1987. – 160 с.: ил.
(В обл.): 60 к.

Рассмотрено практическое использование программируемых микрокалькуляторов типа "Электроника" Б3-34, МК 54 и МК 56 при решении прикладных задач технологии в машиностроении. Книга иллюстрирована примерами с программами решения различных задач из практики производственной и исследовательской деятельности инженера-технолога: размерные расчеты; расчеты при проектировании средств технологического оснащения, математическое моделирование, обработка результатов экспериментальных исследований, технико-экономические расчеты.

Предназначена для широкого круга специалистов различных отраслей машиностроения, использующих в своей практической деятельности микрокалькуляторы.

С $\frac{2704010000-232}{038(01)-87}$ 232-87 ББК 34.5:32.973

©Издательство "Машиностроение", 1987 г.

ПРЕДИСЛОВИЕ

Решения, принятые XXVII съездом КПСС, нацеливают на создание и эффективное использование вычислительной техники и автоматизированных систем в различных сферах хозяйственной деятельности. Самым перспективным направлением развития вычислительной техники является разработка и практическое применение персональных ЭВМ, создание на их базе автоматизированных рабочих мест конструктора, технолога и других специалистов. Работа на них, в свою очередь, требует от инженера не только навыков программирования, но и отличной математической подготовки, квалифицированного обращения с периферийными устройствами ЭВМ, знания заложенных в ЭВМ возможностей. Поэтому сегодня нужно готовить специалистов завтрашнего дня, которые наряду с глубоким знанием специальных предметов будут иметь и компьютерную грамотность. Как просто грамотность, – это не только умение обращаться с пером и бумагой, так и компьютерная грамотность – это не только умение работать с интерфейсом пользователя и умение программировать. Грамотность применительно к персональным ЭВМ – это проникновение в суть процесса вычислений и обработки информации, причем достаточно глубокое, чтобы легко можно было решать различного рода задачи на ЭВМ, так же, как истинно грамотный человек, легко и просто, не задумываясь над правописанием излагает свои мысли на бумаге с помощью пера.

Промышленность может предложить сегодня широкий выбор микрокалькуляторов для научных и инженерных расчетов. Однако программируемые микрокалькуляторы выделяются из всей массы тем, что на ступень ближе к большим ЭВМ и имеют существенно большие возможности для решения широкого круга задач науки и техники. На них можно успешно осваивать программирование, автоматизировать до 95% рутинной вычислительной работы на рабочем месте инженера, с их помощью можно ставить новые, более широкие задачи, повысить качество инженерного проектирования.

Математики и программисты нередко отрицательно отзываются о программируемых микрокалькуляторах, поскольку любая вычислительная задача может быть решена на большой ЭВМ. Однако это мнение неверно. Во-первых, мощные вычислительные машины не отрицают ЭВМ индивидуального пользования для решения частных задач. Во-вторых, работа с большими и средними ЭВМ требует глубокого знания математического обеспечения ЭВМ, специальной подготовки. В-третьих, в работе инженера сейчас существует огромное множество задач, которые невозможно решить с помощью логарифмической линейки или на листке бумаги, но которые еще недостаточно серьезны для того, чтобы считать их на больших ЭВМ. Здесь неоценимую услугу может оказать

программируемый микрокалькулятор, тем более что результат, полученный при расчете, известен только вычислителю. В-четвертых, работа с микрокалькулятором практически не требует времени на подготовку данных и поэтому несравнима с общим временем расчета на вычислительном центре, так как там необходимы перфорация программы и исходных данных, поиск и устранение ошибок, ожидание очереди и т. д. И, наконец, в-пятых, незнание методов решения не позволяет инженеру иногда ставить задачу, не говоря о ее решении.

Микрокалькуляторы обладают языком программирования достаточно высокого уровня, в то же время простым для самостоятельного освоения его инженером. Работа с таким микрокалькулятором стимулирует глубокое освоение математических методов, а следовательно, позволяет поставить и решить более глубокие задачи, может существенно повысить квалификацию инженера в своей специальности и наконец привести к постановке больших задач на ЭВМ. Нельзя отрицать и игрового компонента в работе с микрокалькулятором, а это тоже один из главнейших элементов научно-познавательной деятельности человека.

Существующие в настоящее время работы по программируемым микрокалькуляторам посвящены вычислению ряда широко применяемых функций и решению прикладных задач в области радиотехники или безотносительно к области техники вообще. Применительно к технологии машиностроения постановка и решение задач, выполняемых с помощью программируемых микрокалькуляторов, пока не рассматривались, за исключением задачи размерных цепей, решаемой на микрокалькуляторе модели "Электроника" Б3-21 [26].

В настоящей работе сделана попытка показать возможности постановки и решения задач с помощью программируемого микрокалькулятора в области технологии машиностроения. В основу разработанных программ при решении задач технологии машиностроения положены два принципа: принцип программного анализа ошибок, возникающих вследствие ослабления внимания пользователя или округления микрокалькулятором данных при вычислениях, и принцип программных сообщений пользователю о его дальнейших действиях. Это вызвано тем, что расчеты на микрокалькуляторе выполняет не оператор ЭВМ, а специалист, решающий какую-либо задачу с применением микрокалькулятора. Поэтому важно, чтобы его внимание при проведении продолжительных расчетов было сосредоточено на выполнении других видов работ: анализе результатов, построении графиков, оформлении расчетов и др. В этом случае калькулятор должен работать не просто как вычислитель, а как контролер действий пользователя и партнер в выполнении некоторой программы действий по решению поставленной задачи.

Все описанные алгоритмы могут быть применены на других, более мощных ЭВМ и вычислительных комплексах.

ВВЕДЕНИЕ

Одной из наиболее многочисленных категорий инженерно-технических работников в промышленности являются технологи. Они принимают активное участие в производственном процессе, на всех его этапах. От их работы во многом зависят себестоимость, материалоемкость и качество как разрабатываемой, так и производимой техники.

Повышение производительности, увеличение доли творческого процесса и уменьшение рутинной составляющей труда технолога немыслимы без его автоматизации.

Современный уровень развития средств вычислительной техники позволяет переложить бо́льшую часть черновой вычислительной и оформительской работы на вычислительные комплексы и микроЭВМ.

Рассмотрим основные стадии производственного процесса, в которых используется труд технолога.

1. Анализ технологичности конструкции машин, узлов, агрегатов, деталей. Эта технологическая стадия создания новой техники пока наименее изучена и алгоритмизирована. В настоящее время начинают применять функционально-стоимостный анализ [18]. Эта стадия творческая, вычисления в основном ориентировочные, экономические, поэтому автоматизации данной стадии достаточного внимания пока не уделялось.

2. Систематизация деталей, разработка групповых, типовых и индивидуальных технологических процессов. Эта синтетическая и аналитическая стадия подобна конструкторской разработке новой техники и до некоторой степени может быть алгоритмизирована, а следовательно, автоматизирована.

При проектировании технологических процессов технологу приходится иметь дело с огромным объемом информации: характеристиками материала детали и ее термической обработки; размерами детали и составляющими ее элементами; сведениями о наличии оборудования на данном предприятии, цехе, участке; знаниями о традиционных способах получения и контроля подобных деталей и достижениях в этой области; сведениями об имеющейся в распоряжении технологической оснастке и инструменте; данными о загруженности определенной группы рабочих или станков, где должна изготовляться разрабатываемая деталь; правилами оформления технологической документации, содержащимися в ГОСТах, нормах и стандартах предприятий и др.

Квалификация технолога определяется его способностью оперативно работать со всей этой информацией, учитывать все особенности конкретных технологических условий, проектировать наиболее экономичные оптимальные технологические процессы. Некоторые задачи, стоящие перед технологом, успешно решаются на заводских вычислительных центрах, однако это несложные задачи, связанные с анализом и обработкой большого объема информации, например, по загрузке оборудования участков и цехов, по материальному и трудовому нормированию, подсчету применяемости деталей и т. п. Более сложные расчеты, касающиеся размерного анализа деталей, применения групповой технологии, автоматизированы пока только частично. Поскольку для отдельных задач технологического проектирования не нужно сохранять большие массивы информации, на рабочем месте технолога можно применять программируемый микрокалькулятор.

Что касается автоматизированного оформления технологических процессов с учетом действующих стандартов предприятия на режущий, вспомогательный и мерительный инструмент, то здесь с успехом начинают применять автоматизированные рабочие места технолога на базе вычислительных комплексов

типа "Искра" и др. В этом случае вся информация о действующих операциях с указанием всего необходимого технологического оснащения заносится на внешние носители памяти, например, магнитные ленты или диски. Одновременно с этим в памяти машины формируется библиотека–классификатор технологических операций, созданных в памяти ЭВМ. Технологу остается, пользуясь чертежом детали, в режиме диалога с ЭВМ "собрать" на дисплее технологический процесс из типовых операций, вставить вместо пропусков конкретные размеры. В зависимости от программного обеспечения автоматизированного рабочего места можно проводить сравнения различных вариантов технологических процессов, выбирая наиболее экономичный, проводить размерный анализ, рассчитывать припуски, размеры партий, выбор и загрузку оборудования и многое другое. После окончательной разработки технологического процесса машина может распечатать технологическую документацию с соблюдением всех действующих стандартов предприятия, ГОСТов, без формальных ошибок и аккуратно.

ЭВМ при соответствующем программном обеспечении позволяет избежать грубых ошибок проектирования, таких, например, как отсутствие или неправильное указание размеров, отсутствие требуемых операций или инструментов. Однако поскольку творческое начало остается за человеком, принципиальные ошибки, такие, как неправильный выбор вида заготовки, маршрута обработки могут проникать в технологический процесс.

В настоящее время ведутся работы по многим направлениям автоматизированного проектирования, в том числе по автоматической генерации вариантов обработки и сравнению их эффективности, по разработке новых проблемно-ориентированных языков программирования и языков описания деталей и их поверхностей для ввода в ЭВМ в формальном виде, а не в виде чертежей. Поскольку эта задача выходит за рамки книги, ее в целом касаться не будем. В дальнейшем рассмотрим частные задачи, которые могут быть решены на микрокалькуляторе, но одновременно могут входить в общие задачи, решаемые с помощью систем автоматического проектирования.

Одним из направлений автоматизации труда технолога на этой стадии является автоматическое проектирование технологических процессов обработки деталей на автоматах и станках с ЧПУ. Что касается станков с ЧПУ, здесь выгода очевидна, поскольку технологический процесс (наладка) получается в виде информации, записанной на носителе, используемом в системе числового программного управления станком. Автоматические станки с управлением от распределительного вала (в основном токарно-револьверные автоматы и автоматы продольного точения) требуют достаточно простого и быстрого расчета. Интересно отметить, что в Институте технической кибернетики (ИТК АН БССР) созданы программы для расчета карт наладок на токарные кулачковые автоматы. Однако подготовка информации для расчета на ЭВМ и последующее вычерчивание эскизов в картах наладки, ожидание машинного времени требуют затрат времени не меньше, а больше, чем простой расчет на логарифмической линейке или калькуляторе, вследствие чего эти программы редко используют даже на тех предприятиях, которые имеют свой вычислительный центр.

3. Разработка средств технологического оснащения. Как правило, этой работой занимаются конструкторы по оснастке. Однако главным инициатором и ведущим в выборе принципиальной схемы остается технолог, заказывающий требуемую оснастку. За ним – право выбора основных конструктивных решений. В последнее время в связи с развитием теории и практики конструирования и технологии машиностроения на этой стадии технологу приходится решать различного рода уравнения, вычислять интегралы и проводить другие подобного рода расчеты. Здесь имеются большие возможности для применения ЭВМ и в особенности программируемых микрокалькуляторов.

4. Математическое моделирование. Разрабатывая технологический процесс, технолог должен рассчитать такие параметры обработки, которые бы давали наилучшее качество при наименьших затратах. В большинстве случаев технологические решения, причем не всегда лучшие, подбирают по справочникам, ГОСТам, отраслевым нормалям и стандартам предприятия, в которых отражены производственный опыт и теория. В отдельных случаях технологу приходится прогнозировать некоторые величины, поэтому он должен проводить различные расчеты и эксперименты. В частности, все большее применение на стадии проектирования находит размерный анализ технологических процессов с расчетом размерных цепей. В большинстве случаев его можно проводить на микрокалькуляторе.

Для проведения экспериментальных исследований и внедрения новых технологических процессов на всех крупных предприятиях при отделе главного технолога обычно имеется технологическая исследовательская лаборатория, которая занимается поисковыми работами и внедрением новых технологических процессов. Иногда эксперимент сложен, громоздок, чрезвычайно дорог, например, при испытании какого-либо технологического решения в процессе производства уникального ротора турбины атомной электростанции. В этом случае существенную помощь может оказать математическая модель. В исследовании таких математических моделей реальных процессов ЭВМ незаменимы, а программируемый микрокалькулятор окажет неоценимую услугу при решении и исследовании небольших математических моделей.

5. Статистические исследования. В процессе изготовления и сборки деталей в крупносерийном и массовом производстве значительную роль играют статистические методы управления качеством, которые позволяют минимизировать затраты на технический контроль в производстве и своевременно предупреждать возникновение брака на технологических операциях с одновременным сокращением трудоемкости контроля. Кроме того, при разработке принципиально новых технологических процессов возникает необходимость исследовать точность операции, приспособления и других составляющих элементов производственного процесса. Иногда необходимо выяснить влияние какого-либо фактора на рассеяние размеров детали, используя регрессионный анализ. Здесь также без ЭВМ обойтись невозможно. С помощью программируемого микрокалькулятора можно упростить процедуры математической обработки при статистических исследованиях и уменьшить число ошибок оператора.

6. Экспериментальные исследования. После разработки технологического процесса необходимо внедрить его в производство. На этой стадии часто возникают вопросы, которые невозможно решить без проведения натурных испытаний. В процессе последних результат может получиться в виде осциллограммы,

которая требует математической обработки. Например, во время резания или шлифования возникает вибрация. Математический анализ позволяет выяснить вклад в общую вибрацию элементарных частот, а следовательно, и влияние на нее элементов технологического процесса, с тем чтобы активно воздействовать на образование поверхности. Таким методом анализа может быть гармонический анализ, для которого возможно использование программируемого микрокалькулятора.

7. Технико-экономические расчеты. Подобные расчеты технолог ведет на всех этапах производственного процесса. При расчете мощностей участка цеха и завода, нормировании материальных и трудовых затрат приходится выполнять огромное число повторяющихся достаточно простых вычислений. Такие задачи легко поддаются автоматизации, большинство из них рассчитывается на вычислительных центрах предприятий, но некоторые задачи нужно решать не прибегая к большим ЭВМ. И снова здесь на помощь придут программируемые микрокалькуляторы.

Ниже будут рассмотрены все аспекты автоматизации труда технолога, однако основное внимание уделено применению программируемых микрокалькуляторов "Электроника" Б3-34 и "Электроника" МК 54, МК 56, причем решаться будут как стандартные, широко известные, так и новые задачи для иллюстрации математического подхода к технологии машиностроения и инструментальному производству. Некоторые задачи, такие, как решение дифференциальных уравнений методом Рунге–Кутта, статистические исследования, регрессионный анализ и другие, выполняются на больших ЭВМ; здесь показана возможность их решения на программируемых микрокалькуляторах.

ГЛАВА I.

ПРОГРАММИРОВАНИЕ НА МИКРОКАЛЬКУЛЯТОРАХ

1. ХАРАКТЕРИСТИКА И ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ МИКРОКАЛЬКУЛЯТОРОВ "ЭЛЕКТРОНИКА" Б3-34 И МК 54

Микрокалькуляторы типа "Электроника" предназначены для инженерных и научных расчетов и занимают в ряду ЭВМ промежуточное место между обычными клавишными машинами (ЭКВМ) и большими ЭВМ. По сравнению с обычными ЭКВМ, программируемые микрокалькуляторы имеют следующие преимущества: позволяют быстро ввести программу расчета и выполнить на рабочем месте технолога достаточно сложные вычисления, проведение которых на непрограммируемых калькуляторах громоздко или практически невозможно; обходиться минимальной записью промежуточных результатов, занося их в регистры памяти машины. В сравнении с ЭКВМ программируемые микрокалькуляторы недостатков не имеют, а большим ЭВМ уступают в скорости вычислений, емкости оперативной памяти, функциональных возможностях. Однако проигрыш во времени счета компенсируется отсутствием подготовки исходных данных, ожидания доступа, готовностью к расчетам в любой момент, постоянным присутствием на рабочем месте. Поэтому их наиболее удобно применять при многократных расчетах с достаточно сложными формулами, итерационных вычислениях экстремумов функций и корней трансцендентных уравнений, чис-

ленных методах дифференцирования, интегрирования и решения различного вида уравнений — тригонометрических, дифференциальных и др. Все вычисления ведутся калькулятором автоматически, по заранее составленной и введенной в него программе. Клавиатура микрокалькуляторов "Электроника" Б3-34 и МК 54 состоит из 30 клавиш, назначение которых дано в инструкции для микрокалькулятора [16] и приложении I.

Основные технические данные микрокалькуляторов "Электроника" Б3-34 и МК 54

Диапазон представляемых чисел	$10^{-99} \leqslant x \leqslant 9{,}9999999 \cdot 10^{99}$
Выполняемые операции без программирования	Четыре арифметических действия; возведение в квадрат; извлечение квадратного корня; нахождение обратной величины; вычисление логарифмов $\lg x$ и $\ln x$; вычисление тригонометрических функций $\sin x$, $\cos x$, $\mathrm{tg}\, x$, $\arcsin x$, $\arccos x$, $\mathrm{arctg}\, x$; вычисление экспонент e^x и 10^x; возведение в степень x^y; вызов константы π; изменение знака числа
Выполняемые функции в режиме программируемых вычислений	Команды безусловного перехода; команды перехода по условию $x<0$; $x=0$; $x \geqslant 0$; $x \neq 0$; команды перехода на подпрограмму и возврата из подпрограммы; команды организации циклов; команды косвенного обращения к регистрам памяти; команды косвенных безусловных переходов; команды косвенных переходов по условию $x<0$; $x=0$; $x \geqslant 0$; $x \neq 0$; команда останова программы и индикации содержимого регистра X
Форма представления числа	Естественная с плавающей запятой, $x \in [1 \div 10^8 - 1]$; нормализованная в форме $x = M \times 10^k$, $x \in [10^{-99} \div 1[$; $x \in [10^8 \div 10^{99}]$
Число адресуемых регистров памяти . . .	14
Число регистров стека	5
Число запоминаемых адресов программы	98

Область допустимых значений аргумента и погрешности вычислений:

Функция	*Допустимые значения аргумента*	*Погрешность вычислений*
$\sin x$; $\cos x$	$10^{-49} < \lvert x \rvert < 10^9$	3×10^{-7}
$\mathrm{tg}\, x$	$10^{-99} < \lvert x \rvert < 10^9$	3×10^{-7}
$\arcsin x$; $\arccos x$	$\lvert x \rvert \leqslant 1$	3×10^{-7}
$\mathrm{arctg}\, x$	$10^{-99} \leqslant \lvert x \rvert \leqslant 10^{99}$	3×10^{-7}
x^y	$x > 0$	1×10^{-6}
	Показатель степени результата по абсолютному значению не выше 99	
e^x	$\lvert x \rvert < 230{,}258$	4×10^{-7}
x^2	$\lvert x \rvert < 10^{50}$	1×10^{-7}
10^x	$\lvert x \rvert < 100$	4×10^{-7}
$1/x$	$x \neq 0$	1×10^{-7}
$\sqrt{x}$	$x \geqslant 0$	1×10^{-4}
$\lg x$	$x > 0$	3×10^{-7}
$\ln x$	$x > 0$	3×10^{-7}

Ниже приведено время выполнения различных операций в режиме работы по программе в расчете на одну указанную операцию или переход в секундах.

Ввод цифры; изменение знака числа; очистка регистра X; арифметические действия +; –; ×; ÷; x^2; $\sqrt{x}$; вызов числа π; обмен регистрами стека; кольцевой поворот стека; запись в регистр памяти; команда "Нет операции"	0,2–0,25
Организация циклов; вызов из регистра памяти, из регистра предыдущего результата $X1$; перемещение стека вверх (↑); прямые и косвенные безусловные и условные переходы (в том числе переход на подпрограмму без возврата); косвенная запись в память и вызов из нее	0,3–0,4
Переход на подпрограмму с возвратом из нее	0,4–0,55
Тригонометрические функции $\sin x$, $\cos x$, $\mathrm{tg}\,x$, $\arcsin x$, $\arccos x$, $\mathrm{arctg}\,x$; экспоненциальные e^x, 10^x; логарифмические $\lg x$ $\ln x$	1,2–1,6
Возведение в степень x^y	2,3–2,4

Необходимо учесть, что в зависимости от исполнения микрокалькулятора время вычислений может изменяться на ± 30%.

2. ВВОД И ОТОБРАЖЕНИЕ ЧИСЕЛ

Ввод чисел осуществляют последовательным нажатием клавиш, читая число слева направо, включая десятичную запятую. Знак числа вводят после набора всего числа. Число в калькулятор можно ввести и другим способом. Для этого число предварительно нормализуют, т. е. приводят к виду

$$x = M \times 10^k,$$

где x – число; M – мантисса числа; k – порядок числа.

Тогда после набора мантиссы нажимают клавишу ВП – ввод порядка и вводят порядок числа, причем знак числа вводится до нажатия клавиш ВП, а знак порядка – после нажатия клавиши ВП. Микрокалькулятор работает с нормализованными числами, однако на индикаторе они отображаются как в естественном, так и в нормализованном виде.

Представление числа на индикаторе в нормализованном виде значительно расширяет диапазон работы микрокалькулятора. При вводе в калькулятор слишком больших чисел (более 10^8) или слишком малых (менее 10^{-3}) предпочтение следует отдавать вводу числа в нормализованном виде, поскольку повышается точность расчетов. Например, при вводе числа 0,000987654321 в естественном виде последовательным нажатием клавиш мы сможем ввести только четыре значащих цифры: 0,0009876; при вводе этого же числа в нормализованном виде можно сохранить восемь значащих цифр: 9,8765432 – 04.

3. ВЫПОЛНЕНИЕ ОПЕРАЦИЙ

Различают двухместные операции, которые для своего выполнения требуют двух чисел и занимают соответственно два операционных регистра, и одноместные операции, которые выполняют над одним числом. В программируемых

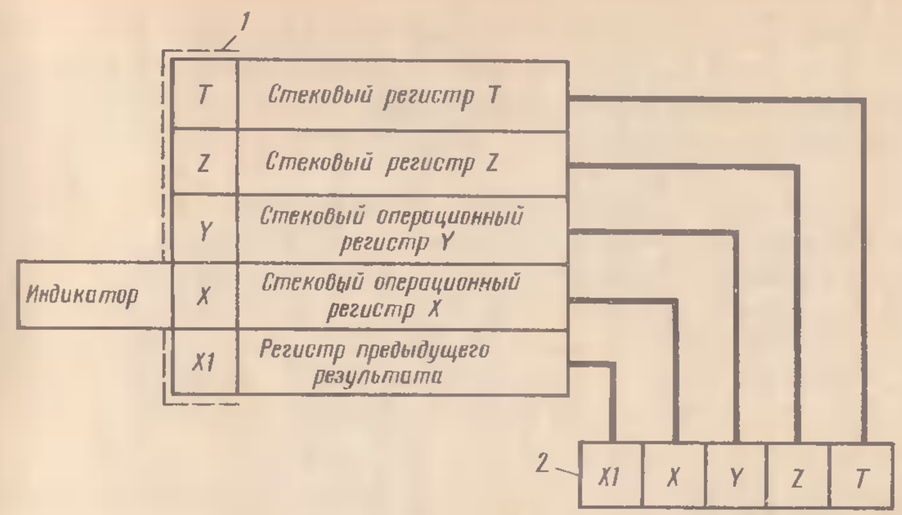


Рис. 1. Схема стековых регистров микрокалькуляторов "Электроника" Б3-34 и МК 54: *1* – вертикальное изображение; *2* – горизонтальное изображение

микрокалькуляторах имеются стековые регистры памяти. В калькуляторах "Электроника" Б3-34, МК 54, МК 56 таких стеков два. Первый из них имеет доступ пользователя, т. е. его можно в случае необходимости просмотреть. Он совмещен с операционными регистрами и автоматически передвигает записанные в него числа при выполнении операций. Второй стек доступа пользователя не имеет; в нем хранятся адреса команд (адресный стек). Он особенно важен для организации последовательного обращения к подпрограммам и выхода из них на адрес, следующий за командой обращения к подпрограмме. В адресном стеке четыре ячейки памяти, поэтому глубина обращения к подпрограммам в этих калькуляторах ограничена числом 4.

Числовой стек в микрокалькуляторах содержит обычно от трех до семи регистров. В микрокалькуляторах "Электроника" Б3-34, МК 54, МК 56 таких регистров пять. На схемах их обычно располагают друг над другом или слева направо (рис. 1). Для удобства обращения и записи их условно называют X, Y, Z, T.

Содержимое регистра X высвечивается на индикаторе. В нем размещен операнд* одноместной операции или один из операндов двухместной операции. Второй операнд двухместной операции находится в операционном регистре стека Y, и на индикаторе его нет, однако в калькуляторах имеется клавиша $\overrightarrow{XY}$, которая позволяет производить обмен содержимым регистров X и Y, тем самым вызывая содержимое регистра Y в регистр X, а следовательно, на индикацию. После нажатия клавиши двухместной операции результат высвечивается на индикаторе и располагается в регистре X. Одновременно с этим содержимое регистра Z размещается в регистре Y, а содержимое регистра T остается в нем

* Операция в ЭВМ – нахождение некоторой величины (элемента данных) в результате выполнения машиной специального действия, указанного командой программы над одной или несколькими исходными величинами. Величины, представляющие объект операции, называют *операндом*. Различают операции обработки данных (вычислительные), операции управления и операции над командами программы (переадресовки).

Таблица 1

Схема перемещения информации в стековых регистрах

Обозначение регистра	Выполняемые операции или вводимые числа											
	A	↑	B	↑	C	+	вх	XY	D	$\sqrt{x}$	÷	↻	x^y
T	–	–	–	–	–	–	–	–	A	A	A	$(B+C)/\sqrt{D}$	$(B+C)/\sqrt{D}$
Z	–	–	–	A	A	–	A	A	C	C	A	A	A
Y	–	A	A	B	B	A	B+C	C	B+C	B+C	C	A	A
X	A	A	B	B	C	B+C	C	B+C	D	$\sqrt{D}$	$(B+C)/\sqrt{D}$	C	C^A
XI	–	–	–	–	–	C	C	C	C	D	$\sqrt{D}$	$(B+C)/\sqrt{D}$	C

и переносится в регистр Z. Таким образом, любая двухместная операция кроме возведения в степень производит сдвиг в регистрах стека на одну ступеньку вниз. Ввод нового числа в операционный регистр X заставляет передвинуться информацию по стеку на одну ступеньку вверх, причем содержимое регистра T пропадает.

Здесь необходимо упомянуть еще об одном регистре памяти – регистре предыдущего результата X1. При выполнении любой операции, исключая операции переходов и обращения к регистрам памяти, содержимое регистра X предварительно засылается в регистр X1 и затем над ним выполняется операция. Поэтому после операции в регистре X1 сохраняется то число, которое до операции было в регистре X. Вызов содержимого регистра X1 в операционный регистр X производится нажатием клавиш $F^{вх}_{\boxed{\uparrow}}$, причем все регистры стека смещаются на одну ступеньку вверх. В дальнейшем, в случае использования клавиш, имеющих несколько значений в зависимости от нажатой префиксной клавиши, будем указывать только на то, которое требуется по смыслу, опуская остальные. При полной занятости адресуемых регистров памяти и в некоторых других случаях удобно пользоваться этим регистром. Перемещения в стеке при выполнении операций и вводе чисел иллюстрирует табл. 1.

В микрокалькуляторах, имеющих стек, как правило, используют обратную бесскобочную запись операций, предложенную польским математиком Яном Лукашевичем. Она заключается в записи двух чисел и после них записи требуемого математического действия, что позволяет сократить запись программ последовательных вычислений и обходиться без скобок. Интересно отметить, что все калькуляторы выполняют одноместные операции в обратной записи.

Обратная запись операций требует разделения двух вводимых операндов с тем, чтобы калькулятор мог отличить один от другого. В этом случае используют команду пересылки (клавиша $\boxed{\uparrow}$), которая осуществляет пересылку содержимого регистров стека на ступеньку вверх и готовит регистр X для ввода нового числа. Автоматическая работа стека позволяет не обозначать конец одного из операндов командой пересылки, если над ним проводится любая одноместная операция в соответствии с задачами расчета или если второй операнд вызывается из регистра памяти.

4. РАБОТА С АДРЕСУЕМЫМИ РЕГИСТРАМИ ПАМЯТИ

Адресуемым называется такой регистр памяти, к которому можно обратиться в любой момент вычислений, нажав клавиши засылки в регистр или вызова из регистра. Для записи числа в адресуемый регистр памяти необходимо это число ввести в регистр X, а следовательно, и на индикацию любым путем: либо совершив какую-либо операцию, либо передвинув информацию в стековом регистре, либо вызвав его из другого регистра памяти, а затем нажать клавишу записи в требуемый регистр и номер регистра. В микрокалькуляторах "Электроника" Б3-34 и МК 56 это клавиши П и №, в микрокалькуляторе "Электроника" МК 54 – X →П и №. Для извлечения числа из регистра памяти необходимо нажать клавишу вызова числа из регистра и номер требуемого ре-

гистра. Для указанных моделей микрокалькуляторов это клавиши ИП и П →X соответственно.

Пример. Вычислить значение $x = 25 \times \sqrt{\sin 33{,}5^\circ}$ и записать это значение в регистры памяти П1 и ПD. Нажимаемые клавиши – 3 3, 5 F sin F $\sqrt{x}$ 2 5 ×. П1 ПD. Индикация результата $x = 18.573115$.

5. ОРГАНИЗАЦИЯ ЦЕПОЧНЫХ ВЫЧИСЛЕНИЙ

Преимущества обратной бесскобочной записи операций и автоматического перемещения информации в стеке легко видны при организации цепочных вычислений. Такие вычисления необходимо проводить, например, при математической обработке материалов эксперимента. Это вычисления, которые описываются формулами вида

$$x = \sum_{i=1}^{n} A_i \cdot B_i \cdot C_i. \qquad (1)$$

Работа с автоматическим стеком при вычислениях по формуле (1) позволяет обойтись без промежуточных записей и сокращает общее число нажимаемых клавишей по сравнению с калькуляторами с арифметической логикой (в них есть клавиша "=").

Последовательность нажатия клавишей и перемещение информации в стековых регистрах показаны в табл. 2.

Таблица 2

Схема перемещения информации в стековых регистрах памяти при выполнении цепочных вычислений

Ввод	Содержимое регистров стека			
	X	Y	Z	T
Число A_1	A_1			
↑	A_1	A_1		
Число B_1	B_1	A_1		
↑	B_1	B_1	A_1	
Число C_1	C_1	B_1	A_1	
×	C_1B_1	A_1		
×	$C_1B_1A_1$			
Число A_2	A_2	$A_1B_1C_1$		
↑	A_2	A_2	$A_1B_1C_1$	
Число B_2	B_2	A_2	$A_1B_1C_1$	
↑	B_2	B_2	A_2	$A_1B_1C_1$
Число C_2	C_2	B_2	A_2	"
×	C_2B_2	A_2	$A_1B_1C_1$	"
×	$A_2B_2C_2$	$A_1B_1C_1$	"	"
+	$A_1B_1C_1+A_2B_2C_2$	"	"	"
Число A_3	A_3	$A_1B_1C_1+A_2B_2C_2$	"	"
↑	A_3	A_3	$A_1B_1C_1+A_2B_2C_2$	"
Число B_3	B_3	A_3	$A_1B_1C_1+A_2B_2C_2$	"
↑	B_3	B_3	A_3	$A_1B_1C_1+$ $+A_2B_2C_2$

Ввод	Содержимое регистров стека			
	X	Y	Z	T
Число C_3	C_3	B_3	A_3	$A_1B_1C_1+$ $A_2B_2C_2$
×	C_3B_3	A_3	$A_1B_1C_1+A_2B_2C_2$	"
×	$C_3B_3A_3$	$A_1B_1C_1+A_2B_2C_2$	"	"
+	$\sum_1^3 A_iB_iC_i$	"	"	"
и т. д.				

6. ОСНОВЫ И ОСОБЕННОСТИ ПРОГРАММИРОВАНИЯ НА МИКРОКАЛЬКУЛЯТОРАХ

Язык программирования – это формальная знаковая система, используемая для связи человека с ЭВМ. Как и всякий язык, язык программирования микрокалькуляторов имеет свою грамматику и семантику. Грамматика определяет строй языка, систему отношений между языковыми категориями, а семантика – значение, смысл конструкций языка.

Для того чтобы составить программу как упорядоченную последовательность действий для ЭВМ, реализующую алгоритм решения некоторой задачи, необходимо знать грамматику и семантику языка программирования и точно их соблюдать. Машина в отличие от человека двусмысленностей не понимает, поэтому в целях устранения непредвиденных ошибок, вызванных неоднозначностью реакций на некоторые неопределенные воздействия, машина выдает сообщение об ошибке.

В отличие от языков высокого уровня язык программирования для калькуляторов содержит выполняемые предложения – операторы. Оператор, так же как и предложение, – единица действия в языке. Например, оператор ИП1 аналогичен предложению. "Число, хранящееся в регистре памяти *RG1*, вывести в операционный регистр X, оставив такое же число в регистре *RG1*, одновременно переслать содержимое операционного регистра X в регистр Y; все регистры стека переместить на ступень вверх". Оператор "+" аналогичен предложению: "Переслать содержимое операционного регистра X в стековый регистр X1, сложить числа, находящиеся в операционных регистрах стека X и Y, результат поместить в регистр X, а содержимое стека переместить на ступеньку вниз".

Операторы вводятся в калькулятор в режиме программирования как последовательные нажатия клавишей (от одной до трех). Операторы можно разделить на три группы:

операторы ввода, с помощью которых осуществляется ввод чисел 0, 1, 2, 3, 4, ..., 9, / –/, ВП;

операторы вычислений, с помощью которых производят операции +, –, ÷, ×, x^2, sin x, x > 0 и т. д.;

служебные операторы, которые предназначены для организации циклов, обращения к памяти, очистки регистра, перемещения в стеке, перехода на подпрограмму и т. д.

Программирование на микрокалькуляторах имеет следующие особенности:

в качестве меток переходов служат адреса операторов (подобно языку БЭЙСИК), поэтому в процессе расчета можно легко изменять метки начала подпрограмм, условных и безусловных переходов, вход в подпрограмму можно выполнять с любого оператора;

условием перехода в операторе перехода по условию является сравнение содержимого операционного регистра X с нулем, поэтому при сравнении двух величин, когда на выходе из условного перехода требуется сохранить одну из сравниваемых величин (обычно она расчетная и только что полученная), следует пользоваться адресуемым регистром памяти, стеком либо регистром предыдущего результата;

оператор организации цикла ставят в конце тела цикла, а параметр цикла всегда равен $i = -1$. Однако с помощью оператора косвенного обращения к памяти через регистр-счетчик цикла, размещенный в цикле, этот параметр можно кратно увеличить; вход в тело цикла можно осуществлять с любого оператора;

условные и безусловные переходы не изменяют содержания стековых регистров, поэтому их можно размещать один за другим;

имеются некоторые запрещенные последовательности операторов, связанные с конструкцией микрокалькулятора (см. прил. III);

возможно использование некоторых команд, которые не оговорены инструкцией, но позволяют расширить словарный запас. Например, команда косвенной адресации КИП ↑ (КП ↑) аналогична команде КИП0 (КП0), но не изменяет содержимого регистра памяти П0 [39], поэтому ее весьма удобно использовать в цикле с регистром-счетчиком, размещенным в П0 (оператор цикла L0);

отсутствие таких операций, как выделение целой и дробной частей числа, абсолютной величины, определение четности, заставляет использовать несколько операторов вместо одного, причем все эти стандартные процедуры должны быть по возможности короче (иллюстрации на примерах в конкретных программах);

сложность наблюдения за ходом выполнения программы при ее отладке заставляет программу отлаживать (а затем и применять) законченными блоками; это относится и к большим ЭВМ;

необходимо учитывать изменение содержимого стековых регистров при выполнении операций; поскольку стек автоматический, в нем постоянно циркулирует информация, которую можно с пользой применять.

Программирование. На первой стадии программирования необходимо составить алгоритм вычислений, другими словами, определить тот необходимый порядок действий, который неизбежно приведет к верному решению задачи. Если вычисления нужно проводить по нескольким сложным формулам последовательно, нет необходимости преобразовывать все формулы в одну. Здесь вполне вероятно возникнут ошибки. В этом случае вычисления по алгоритму проводят последовательно по всем составляющим решение формулам, причем величины, входящие в последующие формулы, должны быть предварительно определены и размещены в регистрах памяти или стека. Для более тщательного изучения составления алгоритмов вычислений необходимо пользоваться источниками [3, 6, 8, 12, 13, 21]. После составления алгоритма определяют ориентировочные области допустимых значений аргументов, ожидаемые результаты. Закрепляют регистры памяти за определенными величинами. Проверяют вычислительный алгоритм на отсутствие повторяющихся участков. Наконец, проверенный алгоритм переводят на входной язык микрокалькулятора — составляют программу в виде последовательности нажатия клавиш.

Основным недостатком программируемых микрокалькуляторов, как отмечалось выше, является ограниченность длины программы (в 98 шагов) и невысокая скорость вычислений. Поэтому для того, чтобы эффективно использовать микрокалькулятор, необходимо создавать такие программы, которые были бы компактны и эффективны. Компактность программы играет решающую роль, поскольку от длины программы зависит, можно ли вообще решать задачу на микрокалькуляторе или отказаться от ее решения в пользу менее сложной задачи, более простого алгоритма ее решения.

Здесь необходимо проявить изобретательность и смекалку, дабы упростить или укоротить программу. Для этих целей рекомендуется широко использовать стековые регистры, подпрограммы, применять адресуемые регистры памяти для временного хранения информации (так называемые буферные регистры) с последующей записью в эти регистры информации, необходимой для вывода, так как вызов из регистра памяти всегда более эффективен, чем повторный расчет в подпрограмме.

Для составления наиболее коротких и эффективных программ необходимо знать прикладную математику и владеть численными методами [6, 8, 12, 13, 21, 23].

При программировании на микрокалькуляторах следует помнить, что вследствие ограниченности разрядной сетки в некоторых случаях нарушается закон коммутативности, т. е.

$$a + b + c \neq b + c + a. \qquad (2)$$

Действительно, если $a = 10^8$, а $b = c = 4$, то при подсчете по левой части формулы (2) получим число a, так как число 4 не умещается в разрядной сетке калькулятора и результат округляется до a. При подсчете по правой части формулы (2) после сложения b и c получим число 8, сумма будет округлена до единицы в последнем разряде, следовательно, результат вычисления по правой части формулы (2) будет 100000010. Больше всего это свойство проявляется при многократных сложениях. Чтобы избежать его, необходимо начинать операции сложения с чисел меньших по абсолютной величине и заканчивать числами большими.

Основной частью сокращения программ является нормирование коэффициентов уравнения, решаемого с помощью данной программы. При нормировании предварительно так изменяют коэффициенты (например, делением), чтобы некоторые из них стали равными единице или по крайней мере целыми числами. Это позволяет высвободить адресуемые регистры и сократить длину программы, так как вызов однозначного числа осуществляется нажатием одной клавиши и занимает всего один шаг в программе.

Например. Исходное уравнение

$$y = A(e^{\lambda x} - 1) + C \cdot x$$

можно преобразовать к виду

$$y_1 = e^{x_1} - 1 + a \cdot x_1$$

подстановкой

$$x_1 = \lambda x;\ a = C/(\lambda \cdot A);\ y_1 = y/A.$$

В результате вместо трех адресуемых регистров памяти, требуемых для первого уравнения, для второго уравнения нужен всего один.

Ввод и редактирование программы. После составления программы ее необходимо ввести в машину. Для этого нажимают клавиши В/О (очистка счетчика шагов и установка его на адрес с кодом 00), F ПРГ. На индикаторе отображаются четыре группы знаков (рис. 2), которые означают (слева направо): I – группа знаков – код оператора последним введенного в память машины в режиме программирования или код последнего выполненного оператора в режиме автоматического счета; II – код предпоследней нажатой клавиши или выполненного оператора в автоматическом режиме; III – код третьей от конца нажатой клавиши или выполненного оператора в режиме автоматических вычислений; IV – номер (адрес) оператора, который предстоит выполнить в автоматическом режиме, или номер оператора, который будет записан в память машины при очередном нажатии клавиш.

При записи программы с нулевого адреса высвечивается только "00" в IV группе знаков. Это означает, что микрокалькулятор готов к записи программы с адреса 00. Однако программу можно записать, начиная с любого, а не только нулевого шага. Это необходимо, например, при изменении формулы в итерационных вычислениях, когда программа приближения остается одной и той же, а изменяется только вид уравнения. При этом перед переходом на программирование нажимают клавиши БП и номер шага, с которого необходимо записывать программу. В таких случаях при записи программ на бумаге важную роль играет адрес команды, записанный слева от обозначения клавиши и возрастающий последовательно от 00 до 97 или до окончания программы. Правильность нажатия клавишей контролируют по индикатору с учетом кода команд (см. прил. II).

Для проверки и редактирования программы в микрокалькуляторах имеются две клавиши ШГ, ШГ, которые позволяют изменить состояние счетчика команд, а следовательно, и индикатора на один шаг за одно нажатие вперед или назад. Для коррекции неправильно введенного оператора необходимо уменьшить содержимое счетчика программы на требуемое число шагов или увеличить его, пока четвертая группа цифр не укажет номер операции, подлежащей коррекции. После этого вводят требуемый оператор, нажимая нужную клавишу или клавиши.

Если контролируемая операция достаточно далеко от индицируемого участка программы, необходимо воспользоваться клавишей безусловного перехода. Для этого микрокалькулятор нажатием клавиш F АВТ переводят в автома-

Рис. 2. Схема отображения на экране информации о программной памяти

тический режим и нажатием клавиш БП и номера шага (например, 01) переводят счетчик команд в требуемое состояние, клавишами F ПРГ переводят в режим программирования и вносят требуемые изменения. При этом счетчик команд регистрирует именно тот номер, который был нажат после клавиши БП. Необходимо заметить, что при ручном наборе цифр и выполнении операции в режиме автоматического счета состояние счетчика команд не изменяется. Оно может измениться только при нажатии клавиши В/О – устанавливается на нуль, ШГ, ШГ – на один шаг за одно нажатие, БП и номера шага – устанавливается такое состояние, которое определяется номером, следующим за клавишей БП. Отсюда следует вывод, что в любом месте программы при ее пошаговом прохождении можно выполнить любое количество операций, а затем приступить к выполнению программы с прерванного места.

Микрокалькуляторы обладают способностью сохранять значение стековых регистров при переходе от режима программирования к режиму автоматической работы, поэтому в любом месте программы можно посмотреть состояние программы, какая операция выполнялась, какая будет выполняться, установить счетчик программы на прежний номер операции, перевести машину в режим автоматической работы и продолжить вычисление по программе.

Линейные программы. Простые, хотя и достаточно длинные вычисления, можно организовать линейными программами. Линейной называется такая программа, операции которой ведутся последовательно от низших адресов к высшим, до конечной операции "останов" (клавиша С/П – стоп/пуск), причем счетчик команд при выполнении каждой операции прибавляет единицу и таким образом осуществляется последовательное выполнение всех операторов программы.

Возможность таких программ ограничена максимальной длиной в 98 шагов.

Разветвляющиеся программы. Как правило, все вычислительные методы, применяемые в технике, построены на итерационных процессах, требующих последовательных приближений, а следовательно, и оценки результата по степени приближения. Такая оценка проводится с помощью логических операторов и в зависимости от выполнения требуемого условия программа может направить вычисления по нескольким ветвям. Кроме того, в целях сокращения длины программы в калькуляторах имеется возможность организовать повторяющиеся участки программы в подпрограммы или циклы, таким образом нарушая последовательное выполнение программы от низшего оператора к высшему.

Разветвляющиеся программы строятся с помощью команд переходов и команд организации циклов. Команды условных переходов могут следовать одна за другой, причем после каждой из них необходимо указать адрес команды перехода.

Команды переходов не изменяют содержимого регистров числового стека, поэтому переход можно осуществлять на адрес, в котором записана любая операция. В языке программирования микрокалькуляторов начало подпрограммы обозначается только адресом первого оператора. Из сказанного следует вывод, что можно начинать подпрограммы с любого адреса и всегда она будет выполняться с этого адреса до команды В/О или адреса условного перехода. Поскольку метки переходов – это тоже адреса операторов и они не локализованы в

подпрограмме, из подпрограммы можно выходить либо безусловно на продолжение вычислений с прерванного адреса, либо по условию на любой адрес программы.

Рассмотрим каждый оператор перехода в отдельности.

Безусловный переход. Безусловный переход реализуется клавишей БП и адресом операции, к которой необходимо перейти (в ФОРТРАНе этому оператору соответствует оператор *GO TO*). В качестве примера рассмотрим фрагмент программы, содержащей команду безусловного перехода.

Адрес	Клавиши	Код операции
42	+	10
43	F x^2	22
44	БП	51
45	81	81
46	ИП2	62
.....	.....	.....
80	–	11
81	+	10
82	ИП2	62
83	÷	13
84	С/П	50

При выполнении этого фрагмента счет будет идти до адреса 44, затем с адреса 46 по 80 операторы будут пропущены и машина перейдет непосредственно к операции, стоящей по адресу 81, выполнит сложение, вызовет из регистра П2 число и после деления на него содержимого регистра X остановится.

Условные переходы. Условный переход, или как его еще называют, переход по условию, реализуется с помощью клавиш $Fx < 0$, $Fx = 0$, $Fx \geqslant 0$, $Fx \neq 0$ и адреса команды, к которой необходимо перейти при невыполнении условия. Здесь необходимо отличать условный переход от перехода *IF* в ФОРТРАНе, который осуществляет переход при выполнении условия. Команды условных переходов обеспечивают выполнение логических операций, смысл которых заключается в проверке содержимого операционного регистра X на выполнение одного из условий: меньше, равно, больше или равно, не равно нулю. Если заданное условие выполняется, калькулятор переходит к выполнению операции, записанной за командой условного перехода и адресом перехода, не принимая его во внимание. Если условие не выполнено, управление передается оператору, адрес которого записан в составе команды условного перехода. В качестве примера рассмотрим фрагмент программы, содержащей команду перехода по условию $x = 0$.

Адрес	Клавиши	Код операции
11	–	11
12	Fx=0	5E
13	25	25
14	Cx	0Г
15	ИП4	64
16	С/П	50
.....	.....	.....

24	ИП5	65
25	9	09
26	9	09
27	С/П	50

В том случае, если к оператору программы с адресом 12 в регистре X будет находиться нуль, машина перейдет к оператору по адресу 14 и на индикаторе высветится содержимое регистра П4, после чего калькулятор остановится. Если к оператору с адресом 12 в регистре X будет число, отличное от нуля, калькулятор перейдет к выполнению оператора, записанного по адресу 25, выведет в регистр X и на индикацию число 99 и остановит цикл вычислений.

Переход на подпрограмму. Такой переход необходим в случае, когда по одним и тем же формулам рассчитывается значение функции с различными значениями одной или нескольких независимых переменных. Переход на подпрограмму реализуется клавишами ПП и номером оператора, с которого начинается подпрограмма. Подпрограмма может быть записана в любом месте программы, начало ее не выделяется ничем, а конец подпрограммы отмечается командой В/О (клавиша "Возврат/Очистка"). Количество подпрограмм в программе не ограничено. Внутри подпрограммы может также стоять команда перехода на подпрограмму. Глубина обращения к подпрограммам ограничена величиной 4, так как адресный стек имеет четыре регистра памяти. Необходимо помнить, что стековые регистры, используемые для счета, и адресный стек расположены в разных функциональных блоках машины и между собой не взаимодействуют, поэтому при переходе на подпрограмму вычислительный стек можно полностью заполнить четырьмя числами и все их передать в подпрограмму, так же как и результат вычисления в подпрограмме, сформированный четырьмя числами стека, полностью передать в вызывающую программу.

Необходимо отметить, что выделение подпрограмм для микрокалькуляторов эффективно в случае выполнения неравенства

$$an > a + 2n + 1,$$

где a – число операторов фрагмента программы; n – число требуемых повторений этого фрагмента.

Еще одна особенность применения подпрограмм состоит в том, что если необходимо повторить несколько раз участки программы, один из которых полностью содержит другой, то подпрограмма должна представлять больший участок, а меньший получится при соответствующем адресовании на требуемый оператор. В качестве примера рассмотрим фрагмент программы.

Адрес	Клавиши	Код операции
18	F$\sqrt{x}$	21
19	ПП	53
20	66	66
21	F$\sqrt{x}$	21
22	+	10
23	ПП	53
24	64	64
25	Fx^2	22

26	–	11
27	ПП	53
28	40	40
29	ПП	53
30	64	64
31	ИП2	62
32	X	12
. . .	. . .	. . .
40	F sin x	1C
41	X	12
42	ИП1	61
43	В/О	52
. . .	. . .	. . .
64	ИП2	62
65	+	10
66	F cos x	1Г
67	ИП3	63
68	ПП	53
69	40	40
70	÷	13
71	В/О	52

В приведенном фрагменте программы результат извлечения корня, полученный в операции по адресу 18, передается в подпрограмму по адресу 66, где после вычисления косинуса и вызова содержимого регистра П3 управление передается в подпрограмму с адреса 40. После выполнения этой подпрограммы микрокалькулятор возвращается к выполнению подпрограммы по адресу 70, и по команде В/О управление передается в основную программу по адресу 21. Далее калькулятор дважды обращается к подпрограмме, но с адреса 64. Имеется и самостоятельное обращение из основной программы к подпрограмме, записанной с адреса 40. В этой подпрограмме операция проводится над двумя операционными регистрами стека, которые формируются в вызывающей программе.

Организация циклов. В калькуляторе имеется команда, позволяющая организовать цикл вычислений. Эта команда используется, когда необходимо вычислить некоторую сумму или произведение с изменяющимся аргументом, определенным образом сформировать стек, причем заранее известное число раз.

Команда организации цикла реализуется клавишами FL0, FL1, FL2, FL3. В таких циклах вычисление идет до команды L0 (либо L1, L2, L3), которая организует вычитание единицы из содержимого счетчика цикла П0 (П1, П2, П3 соответственно) и проводит анализ результата на нуль. Если результат нуль, то в регистре-счетчике сохраняется единица, и управление передается по адресу, следующему за командой организации цикла. Если результат не нуль, то содержимое счетчика уменьшается на единицу, и управление передается оператору, адрес которого указан в команде организации цикла. Этот адрес не обязательно должен быть меньше адреса команды цикла, т. е. начало цикла может

в программе стоять после конца цикла. Это обусловлено тем, что, как и в подпрограммах, в циклах начало не имеет специальной метки, кроме адреса оператора, который может быть адресом и любого другого перехода. В качестве примера рассмотрим фрагмент программы.

Адрес	Клавиши	Код операции
00	ИП1	61
01	ИП2	62
02	Fx^2	22
03	ПП	53
04	48	48
05	–	11
. . .	. . .	. . .
35	FL1	5L
36	00	00
37	ИП5	65
38	С/П	50

Приведенный фрагмент программы показывает, что с адреса 00 по адрес 35 в микрокалькуляторе организован цикл. В зависимости от того, какое число будет записано в регистр памяти П1, этот участок программы повторится столько же раз, а затем управление передается по адресу 37.

С помощью команд условных и безусловных переходов можно обращаться к программе внутрь цикла или выходить из цикла, а также переходить из одного цикла в другой.

Интересная возможность представляется, например, при организации в одной программе двух циклов с использованием одного счетчика и условными переходами из одного цикла в другой. Кроме того, командой организации цикла можно уменьшать содержимое регистра-счетчика без изменения состояния стека, в отличие от команды косвенного обращения, которая также изменяет содержимое счетчика, но вызывает в регистр стека X число.

Адрес	Клавиши	Код операции
11	+	10
12	–	11
13	FL1	5L
14	15	15
15	В/О	52
. . .	. . .	. . .

В этом фрагменте программы операторы с адресами 13 и 14 работают как модификаторы регистра П1. Каждый раз его содержимое уменьшается на единицу без изменения содержания стекового регистра.

В процессе работы в цикле можно обращаться к счетчику цикла для вычисления параметра цикла, отличного от $i = -1$. Для этого используют пересчет по формулам

$$\Delta x = (x_{max} - x_{min})/l,$$

где x_{max}, x_{min} — максимальное и минимальное значение параметра; l — число интервалов, на которые разбит отрезок $[x_{max}; x_{min}]$ и

$$x_i = \Delta x\,(l-i) + x_{min}, \qquad (3)$$

где i — текущее состояние счетчика цикла.

Поскольку текущее состояние счетчика в цикле не становится равным нулю, то значение функции при $x_i = x_{max}$ вычисляют вне тела цикла. Предложенные формулы легко запрограммировать; при этом цикл следует начинать с формулы (3). Однако они занимают много места в программной памяти и требуют увеличенного времени счета. Меньше ресурсов требует вычисление по формуле

$$x_i = x_{i-1} + \Delta x, \qquad (4)$$

но при программировании ее ставят в конце цикла, непосредственно перед оператором организации цикла, при этом в счетчик цикла записывают величину на единицу большую l. Этим можно обеспечить работу в цикле при всех значениях x_i, включая x_{min} и x_{max}.

Косвенное обращение к памяти. Косвенным обращением к регистру памяти называют такое обращение, когда номер адресуемого регистра, в котором записана интересующая нас информация, явно в команде не присутствует. Вызов осуществляется через регистр косвенного обращения, в котором записан номер нужного нам регистра. При этом содержимое регистра косвенного обращения может либо оставаться без изменения, либо определенным образом модифицироваться. В зависимости от регистра косвенного обращения модификация заключается в прибавлении (регистры П4, П5, П6) или вычитании (П0, П1, П2, П3) единицы. Рассмотрим пример с использованием команд косвенной адресации.

Адрес	Клавиши	Код операции
09	ИП1	61
10	×	12
11	КИП0	Г0
12	+	10
13	ИП0	60
14	1	01
15	+	10
16	П0	40
17	XY	14
18	КП0	L0
19	FL2	58
20	02	02
21	С/П	50

При реализации подобного фрагмента программы содержимое операционного регистра X, полученное до оператора по адресу 09, будет умножено на содержимое регистра П1, затем при реализации операции с адресом 11 в зависимости от числа, записанного в регистр П0, будет вызвано содержимое регистра номер которого на единицу меньше числа, записанного в регистре П0. Так, если в регистре П0 записано число 10, в операционный регистр будет вызвано содержимое регистра П9, а число в регистре П0 уменьшится на единицу. Получен-

ная с адреса 12 сумма будет записана в тот же регистр по адресу 18 посредством команды косвенной записи. Операторы с адресами 13–16 нужны, чтобы восстановить номер требуемого регистра. Операторы с адресами 19 и 20 организуют цикл вычислений, начиная с адреса 02.

Косвенное обращение к регистрам памяти удобно использовать при работе с индексированными массивами переменных, когда необходимо обработать массив, состоящий из разных переменных (до 13), по одинаковому алгоритму. Такое обращение осуществляют при обработке больших таблиц с индексированными строками или столбцами.

Команды косвенных переходов. Команды косвенных переходов подобны косвенному обращению к регистрам памяти. Но в этом случае регистр памяти, номер которого входит в команду косвенного перехода, содержит адрес команды перехода. Как и при косвенном обращении к памяти, содержание регистра косвенного перехода может модифицироваться или оставаться неизменным. Поясним сказанное примером.

Адрес	Клавиши	Код операции
00	ИП1	61
01	2	02
02	7	07
03	П4	44
04	XY	14
05	ИП5	65
06	КПП4	–4
07	$F\sqrt{x}$	21
08	С/П	50
09	КИП4	Г4
10	XY	14
11	FL0	5Г
12	06	06
13	2	02
14	$F10^x$	15
15	С/П	50
...	...	...
28	1	01
29	+	10
30	1	01
31	+	10
32	1	01
33	+	10
34	Fx^2	22
35	÷	13
36	В/О	52

Приведенный фрагмент программы работает следующим образом. Число 27 (адреса 01 и 02) записываем в регистр П4, затем командой обмена регистров (XY) в операционный регистр X выводится содержимое регистра П1 и после этого вводится число из регистра П5. Эти два числа в стеке (П5 в регистре

X, П1 в регистре Y) передаются в подпрограмму с 28 адреса (модификация в регистре П4 заключается в прибавлении к содержимому регистра единицы). Закончив выполнение подпрограммы, микрокалькулятор возвращается к адресу 07 программы, где из полученного в результате расчета в подпрограмме числа извлекается квадратный корень и индицируется результат. Повторный пуск микрокалькулятора нажатием клавиши С/П приведет к последующей модификации содержимого регистра П4 (адрес 09). В данной программе этот оператор необходим только для изменения содержимого регистра П4, поэтому после него следует обмен регистрами с тем, чтобы вывести ненужное число из операционного регистра X. Оператор цикла передаст управление оператору с адресом 06, который снова передаст управление на подпрограмму, но уже с адреса 30, и в подпрограмме к содержимому операционного регистра X будет прибавлено на единицу меньше, чем в предыдущий раз. По адресам 28, 30, 32 не обязательно записывается единица, здесь может быть записано любое целое число до 10 или обращение к регистру памяти. Точно так же по адресу 09 вместо модификации адреса начала подпрограммы можно изменить его на любое целое число, при этом существенно усложнить и развить программу.

Аналогичные операции проводятся при косвенных условных и безусловных переходах. Однако необходимо помнить, что при выполнении заданного условия в косвенном условном переходе содержимое адресного регистра не модифицируется, а при невыполнении этого условия – модифицируется.

Редактирование и отладка программ. Случаи, когда программа работает сразу, бывают достаточно редко и при простых программах. Иногда после первого составления и ввода программа "не считает": выдает сообщение о некорректной операции "ЕГГОГ" (от англ. *error* – ошибка), зацикливается или считает неверно. Поэтому всегда после составления и ввода программы в калькулятор желательно пройти программу в пошаговом режиме (нажимая клавишу ПП в режиме автоматического счета). Одновременно с вычислениями, которые производит калькулятор, необходимо выполнить приближенные вычисления в уме. Этим действием проверяется, во-первых, правильность ввода программы, а во-вторых, правильность вычислительного алгоритма. В таком режиме можно контролировать выполняемые калькулятором операции, не нарушая последовательности вычисления по программе в пошаговом режиме. Для этого по мере необходимости переводят микрокалькулятор в режим программирования и по кодам операций определяют выполнение последних. Если при такой проверке ошибок, по мнению оператора, нет, необходимо записать результат и провести снова цикл вычислений с теми же исходными данными, но в автоматическом режиме (клавиши В/О и С/П). После индикации результата его сверяют с результатом, записанным после вычисления в пошаговом режиме счета. Если результаты совпадают, можно продолжить вычисления в автоматическом режиме, если нет, нужно искать в программе некорректные операции (список некорректных операций для микрокалькулятора "Электроника" Б3-34 приведен в прил. III).

Необходимо сделать еще одно замечание относительно счета на микрокалькуляторах. Поскольку сложные тригонометрические показательные функции микрокалькулятор преобразует в простые (сложение и вычитание) путем разложения в ряды, результат всегда имеет погрешность вычислений (см. п. 1

п. 1). В процессе вычислений ошибка не превосходит знака в 5–8-м разрядах, однако иногда при счете из-за ее накопления может появиться выражение arcsin 1,0000003 Естественно, что в данном случае результат должен быть 90°, но калькулятор этого "не знает", поэтому индицирует некорректную операцию. Точно так же ошибка может получиться при логических операциях. Если из значения некоторой сложной функции, равного точно целому числу (например, при возведении целого числа в целочисленную степень), последовательно вычитать единицу, должен получиться нуль. Но, как и в первом случае, результат может отличаться от нуля, и условный переход $x = 0$ никогда не будет выполнен. Об этом надо помнить и "опасные" участки программы составлять так, чтобы избежать ошибок вычисления, например, округляя результат, либо заменяя сравнение с нулем сравнением с некоторой величиной.

Редактирование программы заключается в исправлении неправильно введенных операторов, замене последовательностей операторов, исключении и добавлении новых операторов, замене адресов переходов и т. д. Самым большим неудобством в работе с программируемыми микрокалькуляторами является то обстоятельство, что в случае ввода нового дополнительного оператора остальные операторы программы не сдвигаются автоматически и всю оставшуюся часть программы необходимо набирать вновь. Если же участок программы заменяется другим, более коротким, лишние, ненужные операторы можно заменить оператором КНОП (нет операции), поскольку любое другое состояние программной памяти в этом месте вызовет определенное действие. Естественно, что при последующих работах с этой программой ее необходимо откорректировать и убрать лишние операторы. Например, если мы заменим ненужный оператор нулем или Сх, то в программе будет осуществлен ввод нуля или очистка регистра X. Установить счетчик программной памяти на требуемый оператор можно, используя клавиши безусловного перехода: БП и номер требуемого оператора, либо клавиши сдвига счетчика программы на шаг ШГ, ШГ.

Организация вычислительного процесса. Очень часто при практическом использовании программируемого микрокалькулятора наряду с программированными вычислениями и ручным набором чисел приходится проводить и вычисления, которые нет возможности запрограммировать. Это должно быть заранее определено и расписано, какие действия выполняет микрокалькулятор, а какие необходимо выполнять оператору. Действия оператора тоже можно записать в виде последовательности нажатия клавиш с тем, чтобы затем, не задумываясь, довести эти нажатия до автоматизма.

Необходимо вырабатывать навыки "слепой" работы, не глядя на клавиатуру, однако нужно помнить, что в процессе автоматического счета по программе любая из нажатых клавиш может прервать вычисления, сбив счет. Поэтому "слепой" метод работы относится только к клавиатуре, но правильность ввода и момент окончания вычислений необходимо контролировать по индикатору. Высвечивание результата на индикаторе служит знаком того, что можно выполнять действия или вводить числа, не глядя на клавиатуру, а используя запись порядка нажатия клавишей, таблицу вводимых чисел и т. п.

Нужно стараться по возможности возлагать на калькулятор выполнение наибольшего числа операций, оставляя за собой нажатие только самых необходимых, не требующих внимания клавиш. Если например, рассчитывается и

строится по точкам график некоторой функции, приращение аргумента и переход на начало программы можно запрограммировать, и тогда все действия оператора будут заключаться в нажатии клавиши С/П и записи полученного результата.

При программировании можно использовать калькулятор для диагностирования ошибок ввода чисел с тем, чтобы уменьшить напряжение оператора. Как это делается, будет показано на конкретных примерах.

На рабочем месте оператора должны находиться ручка, чистый лист бумаги для черновика, разграфленный лист бумаги для записи результатов и тетрадь для записи программ. Тетрадь желательно линованная. Все изменения, введенные в программу при редактировании и отладке, должны найти отражение в тетради для записи программ. Полностью выверенные и опробованные на практических расчетах программы желательно переписать начисто, записать название, дату составления, ориентировочное время счета, содержание регистров памяти и стека и результат, некоторые пояснения. После таких записей эта программа превращается в тест и в дальнейшем может быть использована при проверке работоспособности калькулятора. Лист с вычислительными данными необходимо также озаглавить, датировать и записать все исходные данные и их размещение по регистрам памяти.

При построении графиков сложных функций одновременно с вычислениями можно проводить графические построения, тем более что иногда калькулятор будет вычислять значение функции за время от 20 с до нескольких минут. Таким образом сокращаются непроизводительные потери времени на ожидание результата расчета.

Запись программ. Существует несколько видов записи программ для микрокалькуляторов. С одной из них мы уже встретились в процессе изучения особенностей программирования.

Другой вид записи используют при разработке сложных программ или для обучения специалистов, только начинающих освоение микрокалькуляторов. В этом случае программа записывается построчно, с указанием адреса, клавишей, кодов операций, содержания регистра стека, иногда с указанием содержания регистров косвенной адресации и счетчиков цикла и, наконец, комментариев к выполняемым операциям. Такая запись в наибольшей степени объясняет и комментирует программу и позволяет через продолжительные периоды времени быстро включаться в работу с ней. К недостаткам записи этого вида следует отнести громоздкость и излишнюю информативность программы после понимания сущности ее работы.

И, наконец, третий вид записи, предложенный Я.К. Трохименко и Ф.Д. Любичем [36], является самым компактным. Записывают по 10 операторов в строке, с указанием только функции (оператора), но не нажимаемых клавиш. Такая запись позволяет исключить всю лишнюю информацию и представлять программы в очень сжатом виде. Адрес команды благодаря такой записи читается очень просто, каждая строка начинается с адреса, оканчивающегося нулем, а заканчивается адресом, последняя цифра которого – девять. Первая строка имеет адреса с числом десятков 0, вторая – 1 и т.д. Все адреса по первому столбцу будут иметь номера, кратные 10 :00; 10; 20; ...

К недостаткам такого вида записи следует отнести не всегда ясное представление пользователя о работе программы, что не позволяет в полной мере осуществить все замыслы программиста, сложность переложения прикладных программ, если исходную программу необходимо приспособить к решению родственной задачи, и сложность в исправлении опечаток, вкравшихся при издании программ. В случае записи как операторов, так и их кодов опечатки исправляются достаточно просто.

В настоящей работе в целях экономии места использована компактная запись программы, но при этом детально разъяснены алгоритмы работы программы и даны подробные комментарии к ним.

ГЛАВА II.
РАЗМЕРНЫЕ РАСЧЕТЫ ПРИ ПРОЕКТИРОВАНИИ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ

1. РАЗМЕРНЫЕ РАСЧЕТЫ ДЕТАЛЕЙ ПРИ ГРУППОВОЙ ТЕХНОЛОГИИ

Значение групповой технологии в последнее время все возрастает в связи с ориентацией машиностроения и приборостроения на гибкие, переналаживаемые производства. Здесь огромную роль играет сокращение трудоемкости подготовки производства новых изделий, поскольку этот этап занимает наибольшее время в разработке и внедрении новой техники. В условиях мелко- и среднесерийного производства небольшие детали (диаметр 3–12 мм, длина 5–15 мм) в количествах от 1 до 20 тыс. штук в год изготовляют на токарных кулачковых автоматах. Выгоды от изготовления мелких деталей на автоматах всегда выше, чем от изготовления их на универсальных токарных станках, вследствие более высокой точности и низкой стоимости. Однако из-за широкой номенклатуры деталей и большого количества кулачков без какой-либо систематизации даже на подобные детали иногда проектируют и изготовляют индивидуальную оснастку. Токарные автоматы позволяют осуществлять определенные регулировки и комплектовать кулачки таким образом, что становится возможным обработка на одной оснастке целой группы деталей.

Рассмотрим, например, автоматы продольного точения (АПТ). В них передачу движения от кулачка распределительного вала к исполнительному механизму осуществляют посредством передаточного механизма с изменяющимся передаточным отношением плеч рычагов. Общая длина детали, обработанная на одном кулачке, может быть изменена в 3 раза; при этом пропорционально изменяются длины всех ступеней детали. Одновременно с изменением передаточного отношения можно менять относительные положения суппортов режущих инструментов, варьируя тем самым длину резания.

Таким образом, используя совместно две регулировки, можно увеличить длину детали, оставив без изменения или даже уменьшив длину некоторых ступеней детали, причем можно обрабатывать детали геометрически неподобные, с использованием одного комплекта кулачков.

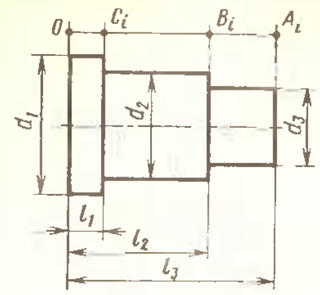


Рис. 3. Деталь–представитель группы шарнирн штифтов и винтов, обрабатываемых на автомат продольного точения по групповой технологии

Дадим экономическое обоснование изг товления деталей типа шарнирных штифт (рис. 3) на АПТ по групповой технологи Сравним только переменные затраты на изг товление деталей. Экономически выгодно изг товлять оснастку для партии деталей n при выполнении неравенства

$$З_а n + З_{осн} \leqslant З_у n,$$

откуда

$$n_{min} \geqslant \frac{З_{осн}}{З_у - З_а},$$

где $З_а$ – затраты на автоматную операцию, коп.; $З_{осн}$ – затраты на изготовл ние оснастки, коп.; $З_у$ – затраты на операцию, выполняемую на универсальн оборудовании.

Для рассматриваемых деталей можно принять, что $З_а = 5$ коп., $З_{осн}$ $= 6000$ коп., $З_у = (5 \div 10) \times З_а = 25$ коп. Тогда минимальная партия детал для которой выгодна индивидуальная наладка, составит $n = 3000$ детал В случае групповой наладки возрастают холостые ходы из-за увеличения вре ния, поэтому затраты на групповую автоматную операцию на 10–30% вы чем при индивидуальной. С учетом того, что на одной групповой наладке об батывают N групп деталей, условие экономичности обработки будет

$$З_{а.г} n + З_{осн}/N \leqslant З_а n + З_{осн},$$

откуда

$$n_{max} \leqslant \frac{З_{осн}(1 - 1/N)}{З_{а.г} - З_а} \approx \frac{З_{осн}}{З_а} \frac{(1 - 1/N)}{0{,}3},$$

где $З_{а.г}$ – затраты на автоматную операцию при групповой обработке.

Экономически выгоднее обрабатывать детали при групповой наладке сравнению с обработкой на универсальном оборудовании при величине парти

$$З_{а.г} n + З_{осн}/N \leqslant З_у n,$$

откуда

$$n \geqslant \frac{З_{осн}}{N(З_у - З_{а.г})}.$$

При $N = 10$ и исходных данных, принятых выше, экономически выгод групповая автоматная технология, по сравнению с индивидуальной автомат или универсальной при размере партии не более $n \leqslant 3600$ и не менее $n \geqslant 30$ талей.

Таким образом, при средних исходных данных экономически целесообр ная партия деталей для групповой обработки составляет 30–4000 шт.

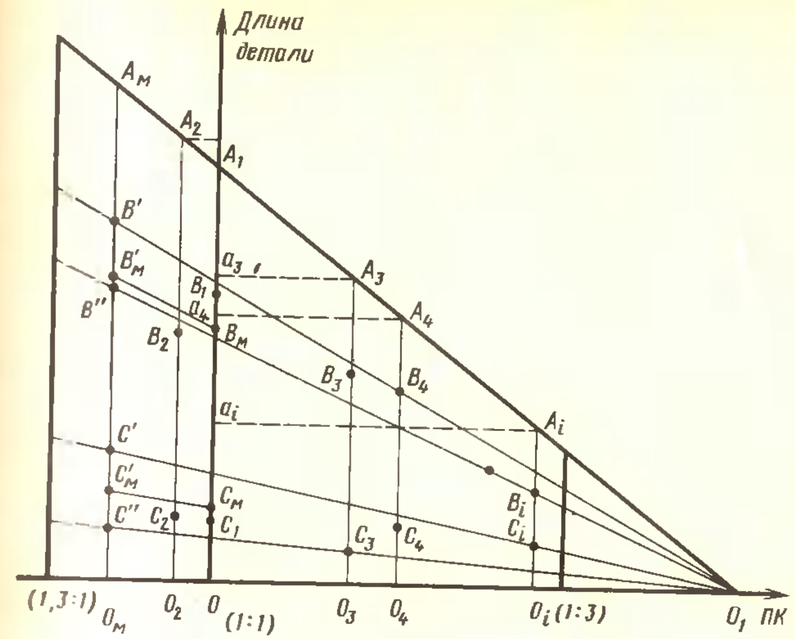


Рис. 4. График передаточных коэффициентов и размещение на нем координат деталей

Метод графического анализа при групповой обработке цилиндрических талей на АПТ описан в работе [30]. Суть его заключается в следующем. Де ли предварительно группируют в зависимости от возможности обработки на ной модели автомата. Группировка деталей по принципу подобия подразу вает подбор деталей исходя из одинакового числа ступеней обработки. За ы из всей группы выбирают деталь, имеющую наибольшую общую длину, с м, чтобы остальные можно было бы получить простым уменьшением переда чного коэффициента шпиндельной бабки (ПК ШБ).

Строят график передаточных коэффициентов, который представляет со й линию, наклонную к оси абсцисс, пересекающую ось ординат в точке, рав й максимальной длине наибольшей детали в группе. Расстояние по оси абс сс от начала координат до точки пересечения с графиком передаточных коэф циентов разбивают на три части и проводят вертикаль, соответствующую пе даточному коэффициенту, равному 1/3 (рис. 4). Ось абсцисс соответствует ередаточным коэффициентам, ось ординат – длине деталей или их ступеней.

Так как передаточный коэффициент шпиндельной бабки может изменять в пределах 1–1/3, то и все детали должны попасть в этот интервал между ью ординат и вертикальной прямой, соответствующей этому ПК. Затем каж ю деталь анализируют индивидуально.

Общую длину некоторой детали в масштабе графика располагают таким разом, чтобы при вертикальном расположении она касалась оси абсцисс и гра ика передаточных коэффициентов. На ней точками отмечают длины всех сту ней. После обработки всех деталей из точки пересечения графика переда чных коэффициентов с осью абсцисс проводят линии таким образом, чтобы и охватывали минимальные и максимальные координаты всех ступеней (точ-

ки B_i и C_i). Определяют возможность регулирования с помощью поперечны суппортов сравнением максимального регулировочного перемещения суппор с максимальным размахом координат одной ступени. Если размах этот боль пределов регулирования, исключают деталь, определяющую размах координ ступени большей величины регулирования.

После подобного анализа остается группа деталей, которая может быть и готовлена с помощью одного комплекта кулачков при одной наладке автом та, причем переналадка занимает минимум времени. Теперь необходимо реш вопрос о том, на какую деталь проектировать кулачки с тем, чтобы можно б ло обработать все детали данной группы.

Анализ показывает, что среди реальных деталей такой детали, как пра ло, не бывает. Необходимо разработать деталь-макет. Эта деталь-макет дол иметь максимальные длины ступеней, подлежащих обработке, и минимальн длины ступеней, не подлежащих обработке. Как правило, не подлежит обраб ке последняя ступень детали, так как на токарных автоматах используют кал рованные прутки. Следовательно, деталь-макет можно построить, приведя к передаточному коэффициенту ПК=1 деталь, у которой третья ступень минима ная, а длина второй ступени равна расстоянию от минимальной координат третьей ступени до максимальной координаты второй. Первая ступень — расстояние от минимальной координаты второй ступени до максимальной ко динаты первой (максимальной длины детали). Таков метод размерного а лиза деталей и синтеза детали-макета при проектировании групповой техно гии обработки деталей на автоматах продольного точения. Его достаточно пр то преобразовать в вычислительный алгоритм, тем более что графические по роения требуют аккуратности, точности и несколько громоздки.

Использование программируемого микрокалькулятора поможет упр тить и автоматизировать вычисления.

Алгорим и блок-схема расчета.

1. Выбрать из группы подобных деталей (см. рис. 3) деталь максималь длины и ввести в память машины длины всех ее ступеней. Координаты A, B соответствуют регистрам памяти ПА, ПВ, ПС. Перейти к п. 2.

2. Ввести в регистры памяти машины длины всех ступеней и общую дли i-й детали. П7 — координата B_i, П8 — координата C_i. Регистр X (индикатор) общая длина детали A_i. Перейти к п. 3.

3. Вычислить передаточный коэффициент шпиндельной бабки для i-й дета ПК$_i$: A_i/A_{max} = ПК$_i$. Перейти к п. 4.

4. Если ПК$_i$ меньше 1/3, перейти к п. 5; если больше или равно, то к п. 6.

5. Индицировать невозможность обработки детали на данной наладке (вы вечивание 0). Перейти к п. 2.

6. Извлечь из регистра П7 координаты B_i, привести ее к ПК=1 делением ПК$_i$: B_i/ПК$_i$ = B_i'. Перейти к п. 7.

7. Сравнить приведенное значение B_i' с максимальным B_{max}, хранимым регистре ПB. Если приведенная координата меньше максимальной, перейти п. 11; если больше или равна, то к п. 8.

8. Вычислить разницу между приведенным значением B_i' и минимальны B_{min}, хранимым в регистре П1. Умножить ее на ПК$_i$. Перейти к п. 9.

9. Сравнить приведенную разность с максимальной величиной регулирова- ния поперечных суппортов. Если разность меньше или равна величине регули- рования, записать B_i' в регистр ПВ в качестве B_{max}, перейти к п. 14; если боль- ше, то к п. 10.

10. Индицировать невозможность обработки данной детали по причине слишком большой длины ступени (высвечивание 99). Перейти к п. 2.

11. Сравнить приведенную координату B_i' с минимальным значением B_{min}, хранимым в регистре П1. Если приведенная координата меньше минимальной, перейти к п. 12; если больше или равна, то к п. 14;

12. Вычислить разницу между приведенной координатой B_i' и максималь- ным значением B_{max} Умножить ее на ПК$_i$. Перейти к п. 13.

13. Сравнить приведенную разность с максимальной величиной регулирова- ния поперечных суппортов. Если разность меньше или равна величине регули- рования, записать B_i' в регистр П1 в качестве B_{min}, перейти к п. 14; если боль- ше, то к п. 10.

14—21. Повторить содержание пп. 6—13 для меньшей ступени детали C_i. Перейти к п. 22.

22. Увеличить содержимое счетчика деталей на единицу. Перейти к п. 2.

После выполнения программы, построенной по этому алгоритму, в регист- рах памяти ПА, ПВ, ПС будут сформированы максимальные координаты соот- ветственных ступеней деталей, а в регистрах П1 и П3 соответственно координа- ты B_{min} и C_{min} минимального значения. Блок-схема алгоритма метода пред- ставлена на рис. 5. В процессе вычисления автоматически отбрасываются те де- тали, которые не могут быть обработаны при такой наладке, об этом сигнали- зирует микрокалькулятор (высвечивая либо 0, либо 99). Поэтому оставшиеся детали необходимо снова сгруппировать и обработать по предлагаемому ал- горитму.

В случае, когда координата B_i ступени детали прошла испытания, ее макси- мальное или минимальное значение будет записано в соответствующий регистр памяти. Если в дальнейшем координата C_i не выдержит испытания, то появится сигнал о непригодности детали, хотя одна из ее координат уже записана; это яв- ляется недостатком данного алгоритма.

Нужно сказать, что такой случай чрезвычайно редкий, поскольку разброс оси головки штифта или винта всегда существенно меньше разброса его шар- нирной части (ступень B_i), поэтому на работу этот недостаток алгоритма не влияет.

Расчет детали-макета. Из деталей, которые отработаны на микрокалькуля- торе, необходимо сформировать деталь-макет. Эти вычисления достаточно прос- тые и неповторяющиеся, поэтому их можно провести в ручном режиме. Для этого необходимо из содержимого регистра ПС (максимальное значение C_{max}) вычесть регулировочное перемещение поперечного суппорта (регистр П5); поскольку данная ступень по условию не обрабатывается, ее можно получить простым сдвигом суппорта. Если получится отрицательная величина, принять координату этой ступени равной нулю, а если положительная величина, то дан- ная разность будет координатой третьей ступени детали-макета. К этой коор- динате третьей ступени детали-макета следует прибавить разность между содер-

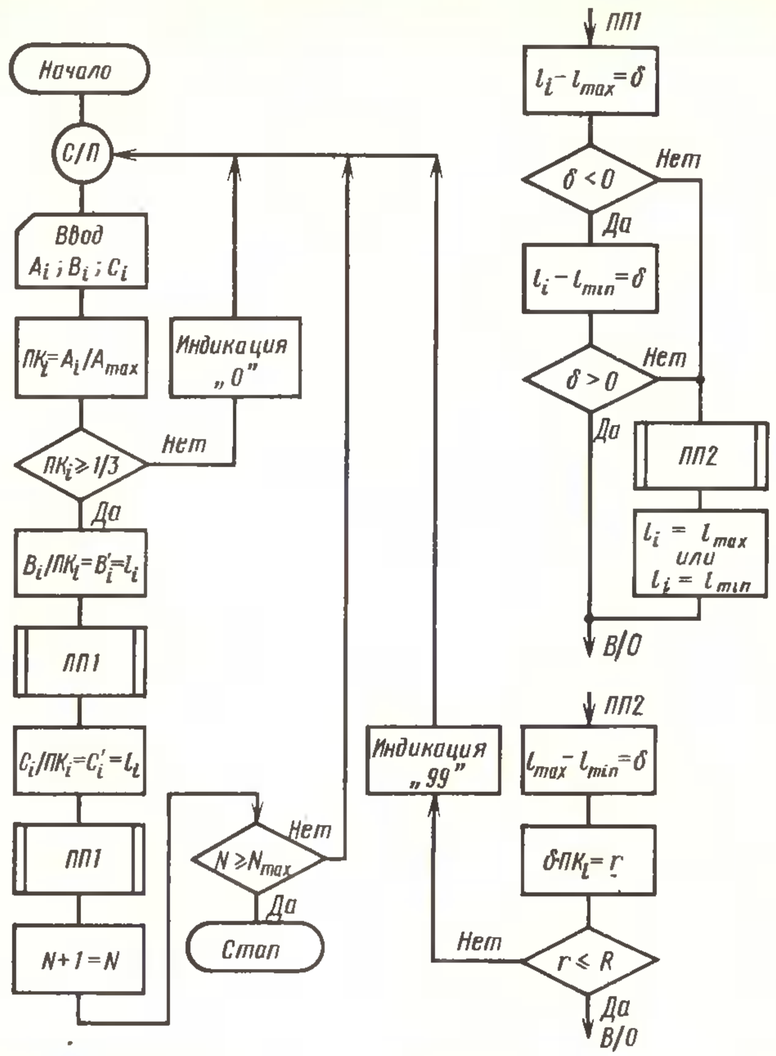


Рис. 5. Блок-схемы алгоритма размерного анализа деталей и синтеза детали-макета

жанием регистров ПВ и П3 (разность между максимальным значением коор-динаты второй ступени и минимальным значением третьей – максимальная дл на обработки второй ступени, которая должна быть предусмотрена при проек-тировании кулачка).

Так получается координата второй ступени детали-макета. К этой коорди-нате следует прибавить разность между регистрами ПА и П1 (разность межд максимальной длиной детали и минимальной координатой второй ступени – максимальная длина обработки первой ступени детали). Если полученная сум-ма отличается от регистра ПА (что обычно и бывает), необходимо эту сумм разделить на содержимое регистра ПА, после чего получим ПК для детали-ма кета.

Если ранее полученные координаты второй и третьей ступеней разделить н ПК детали-макета, получим координаты детали-макета, приведенные к длин

наибольшей детали в группе. Это действие необходимо, поскольку, если деталь-макет получится длиннее наибольшей детали в группе, будет невозможно обра-ботать на данной оснастке наиболее короткие детали.

Программа 1/34. Автоматический анализ размеров деталей и подготовка детали-макета при проектировании групповой наладки на автоматы продольно-го точения:

С/П	ИПА	÷	П0	3	1/x	–	x<0	12	0
БП	00	ИПВ	П9	ИП1	П4	ИП7	ПП	38	ПВ
ИП4	П1	ИПС	П9	ИП3	П4	ИП8	ПП	38	ПС
ИП4	П3	ИП6	1	+	П6	БП	00	ИП0	÷
П7	ИП9	–	x≥0	52	ИП7	ИП4	ПП	64	П9
БП	62	ИП7	ИП4	–	x<0	62	ИП9	ИП7	ПП
64	П4	ИП9	В/О	–	ИП0	X	ИП5	XY	–
x<0	76	9	9	БП	00	ИП7	В/О		

Содержание регистров памяти:

П1, ПВ – B_{max} – координата средней ступени наиболее длинной детали в группе;

П3, ПС – C_{max} – координата последней ступени детали (последней в смысле очередности обработки, с наибольшим диаметром), имеющей наибольшую общую длину;

П7, П8 – координаты B_i и C_i соответственно;

П5 – максимальное значение осевого регулировочного перемещения суп-порта;

П6 – счетчик отобранных деталей; он же индицирует сигнал об окончании анализа размеров детали и отработке ее координат, перед началом счета необ-ходимо записать нуль;

ПА – общая длина наибольшей детали в группе;

П0, П4, П9 – буферные регистры памяти, используемые в подпрограмме для хранения промежуточных результатов вычислений;

П2 и ПD не используются.

Работа с программой. Перед началом работы по программе необходимо отобрать детали, на которые нужно разработать карту наладки, рассортировать их по диаметрам и максимальной длине с тем, чтобы все они могли быть изго-товлены на одной модели автомата. Затем выбрать самую длинную из всех де-талей и ее общую длину (в миллиметрах) заложить в регистр памяти ПА. Ко-ординату средней ступени B заложить в регистры ПВ и П1, координату корот-кой ступени – в ПС и П3. Затем остальные детали расположить в любом поряд-ке и информацию о первой из них записать в память машины: координату сред-ней ступени B_i – в регистр П7, координату короткой ступени C_i – в П8, а об-щую длину ввести в регистр X. Первый раз нажать клавиши В/О и дважды С/П, последующие только один раз С/П.

Если в результате нажатия клавиши спустя 15–25 с высветится цифра 1, а при последующих нажатиях 2, 3 и т. д., значит, микрокалькулятор отработал размеры детали, она годится для обработки при групповой наладке.

Одновременно с этим в регистрах ПВ и П1 отсортировываются соответст-венно максимальная и минимальная координаты ступени B, а в регистре ПС и

ПЗ – ступени *С* обработанной группы деталей. Причем сортировка идет тольк в том случае, если разница между максимальной и минимальной координата ми, приведенная к фактической длине *i*-й детали, не превосходит величину р гулирования суппортов. В противном случае на индикаторе высвечивается чи ло 99, означающее, что деталь не принята в наладку из-за слишком большой и малой координаты какой-либо ступени, и это обстоятельство не позволит вкл чить данную деталь в разрабатываемую карту групповой наладки. Координат ступеней этой детали не заносятся в регистры, хранящие максимальные и ми нимальные размеры координат ступеней. Если деталь не подходит в групповую наладку в связи с тем, что имеет слишком короткую общую длину, на экран высвечивается 0, означающий, что передаточный коэффициент шпиндельно бабки получается менее 1/3. Не включенные в группу микрокалькуляторо детали необходимо отмечать, с тем чтобы потом из них составить новую группу

2. ОРГАНИЗАЦИЯ УЧЕТА ДЕТАЛЕЙ, ОБРАБАТЫВАЕМЫХ НА ТОКАРНЫХ АВТОМАТАХ И ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ ОБРАБОТКИ ВНОВЬ РАЗРАБАТЫВАЕМЫХ ДЕТАЛЕЙ НА ИМЕЮЩЕЙСЯ ОСНАСТКЕ

Специфика групповой обработки деталей на токарных автоматах состоит том, что в отличие от других видов автоматического оборудования (зубофре зерные, копировально-фрезерные, сверлильные и резьбошлифовальные полуа томаты и автоматы) на одной оснастке можно обработать большое число типо размеров деталей.

Групповая обработка открывает перед технологом новые возможности части ускоренной технологической подготовки и изготовления оснастки дл вновь разрабатываемых деталей. Ведь если деталь полностью подходит для об работки при данной групповой наладке, смысл разработки технологическог процесса заключается во внесении обозначения данной детали в групповой тех нологический процесс, и эту деталь можно непосредственно изготовлять.

Однако групповая технология требует тщательного учета разработанны наладок и технологической оснастки. Такой же учет нужен и при индивидуаль ном проектировании технологических процессов, чтобы не существовали на ладки-близнецы и одинаковая оснастка под разными шифрами. Учет разрабо та́нных технологических процессов и применяемой оснастки даже при индиви дальном проектировании в конечном итоге приводит к групповым методам обработки.

Учет наладок можно вести в журнале или канцелярской книге, выделив дл каждой группы деталей двойной лист. На левом развороте в правом верхне углу следует начертить эскиз детали с обозначением всех размеров условным буквами, ниже его поместить таблицу с указанием конструкторского обозна чения номера детали, конкретных исполнительных размеров, материала, годо вого количества, шифра наладки с указанием степени унифицированности тех нологической операции (единичная, групповая и т. д.). При аккуратном запол нении на все эти сведения для одной детали достаточно одной строки. На пра вом развороте листа в верхней его части следует поместить маршрутное описа ние технологического процесса для всей группы деталей данного листа.

Естественно, типовое маршрутное описание должно быть составлено с уче всех операций, требующихся для изготовления наиболее сложной детали группы. Для остальных деталей маршрутные описания могут быть получены ис ключением лишних операций. При этом, поскольку мы учитываем наладки, а сами детали, технологические процессы могут существенно отличаться. Существенным условием является подобность деталей на автоматной обработке.

И, наконец, нижняя половина правого разворота должна содержать сведе о всей стандартной и специальной технологической оснастке.

Если токарно-револьверные автоматы имеют нерегулируемые механизмы передачи движения от распределительного вала с кулачками к исполнительному органу и требуют простого учета деталей с одинаковыми длинами обработан ных поверхностей, причем общую длину можно не учитывать, то автоматы про дольного точения, как указывалось в предыдущем параграфе, в этом смысле имеют несравненно бо́льшие возможности. Поэтому, имея под рукой журнал учета наладок, не всегда сразу можно сказать о пригодности вновь поступившей детали данной группы для обработки на автомате продольного точения. Необходимо привести деталь с помощью передаточного коэффициента к длине де тали на проверяемой оснастке, определить уменьшение всех ее ступеней. Опре делить возможности регулирования суппортов и т. д. Расчеты несложные, но их слишком много для одной карты наладки и очень много становится, если де тали необходимо проверить по нескольким подобным картам наладок. Поэто му в таких случаях становится быстрее и проще разработать индивидуаль ную наладку на вновь поступившую деталь, нежели подобного рода расчеты. Использование ЭВМ позволяет технологу значительно облегчить задачу. Так, применение программируемых микрокалькуляторов дает возможность 3–5 мин ввода программы и данных и 10–15 с расчета без всякого напряже ния внимания проводить такие проверки.

Алгоритм автоматизированного исследования пригодности детали для обработки на данной наладке (рис. 6).

1. Определить передаточный коэффициент делением длины исследуемой детали на длину максимальной детали группы, либо на длину детали-макета, либо на значение подъема на кулачке шпиндельной бабки. Перейти к п. 2.

2. Сравнить вычисленное значение ПК с величиной 1/3 (максимально возможное регулирование ПК). Если ПК $<$ 1/3, перейти к п. 3; если ПК $\geqslant$ 1/3, к п. 4.

3. Индицировать невозможность обработки на данной наладке в связи с недостаточной длиной детали (высвечивание 0). Вернуться к п. 1.

4. Определить приведенную длину средней ступени детали, разделив ее на ПК. Перейти к п. 5.

5. Вычислить разность между приведенной длиной средней ступени детали и длиной соответствующей ступени детали-макета. Умножить эту разность на ПК. Если разность меньше нуля, перейти к п. 7, иначе – к п. 6.

6. Если разность, с учетом ПК детали меньше или равна величине регулирования поперечного суппорта, перейти к п. 8; если нет, то к п. 7.

7. Индицировать невозможность обработки детали на исследуемой оснаст-

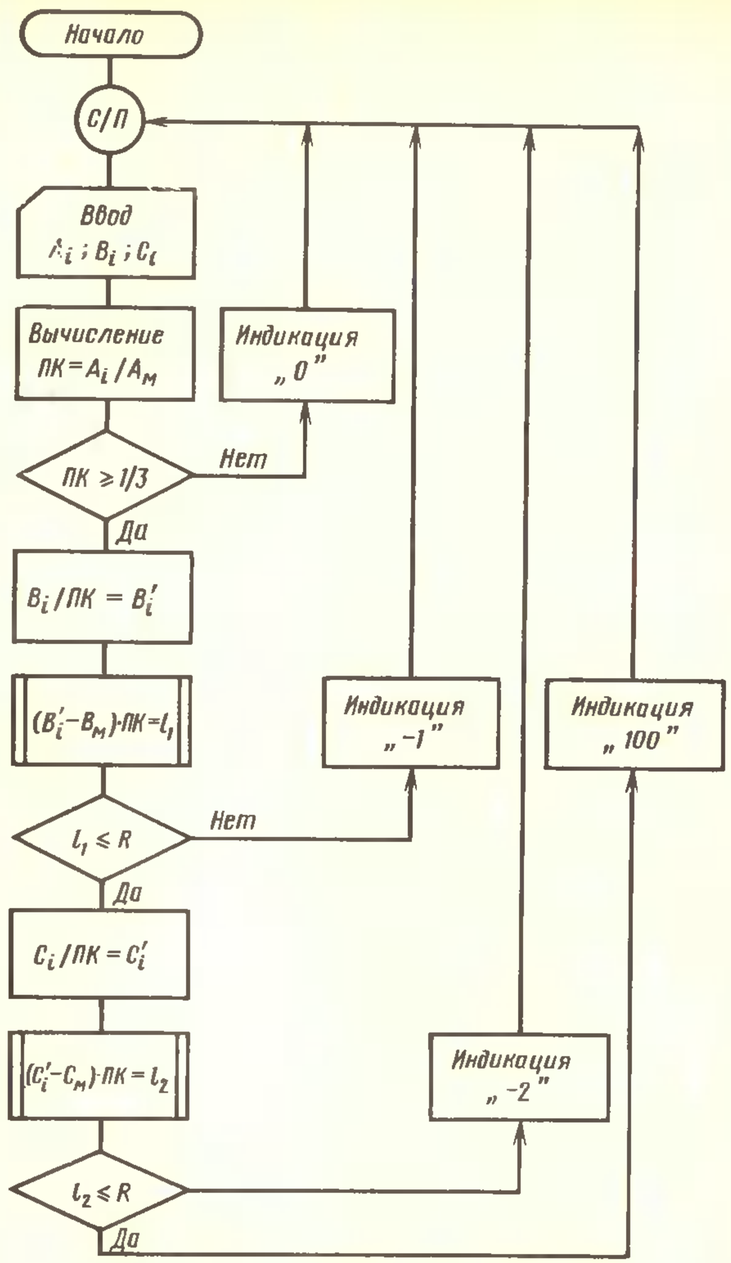


Рис. 6. Блок-схема алгоритма автоматизированного исследования пригодности детали д обработки на групповой оснастке

ке в связи с очень большой или малой длиной средней ступени (высвечиван −1). Вернуться к п. 1.

8. Определить приведенную длину меньшей ступени детали, разделив ее ПК. Перейти к п. 9.

9. Вычислить разность между приведенной длиной меньшей ступени и соо ветствующей длиной ступени детали-макета. Умножить эту разность на ПК, т

ными приведя к натуральной длине исследуемой детали. Если разность меньше ля, перейти к п. 11, иначе – к п. 10.

10. Если полученная разность меньше или равна величине регулирования поперечного суппорта, перейти к п. 12; если нет, то к п. 11.

11. Индицировать невозможность обработки детали на данной наладке в ви с очень большой длиной меньшей ступени (высвечивание – 2). Вернуться п. 1.

12. Индицировать пригодность детали к обработке на данной наладке (высвечивание 100). Вернуться к п. 1.

Программа 2/34. Автоматизированное исследование пригодности детали ля обработки на групповой наладке:

С/П	ИПА	÷	П0	3	1/x	–	x<0	12	0
БП	00	ИП7	ИП0	÷	ИПВ	ПП	41	x≥0	24
1	/–/	БП	00	ИП8	ИП0	÷	ИПС	ПП	41
x≥0	36	2	/–/	БП	00	1	0	0	БП
00	–	x≥0	20	ИП0	X	ИП5	–	В/О	

Содержание регистров памяти:

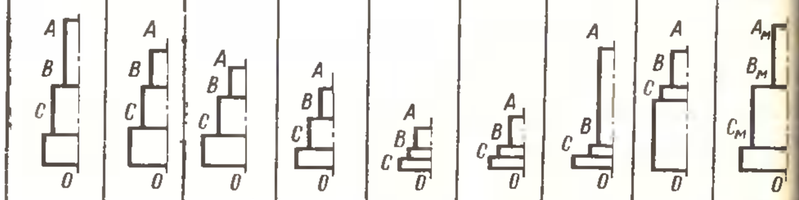
А – общая длина детали-макета, подъем на кулачке $A_м$;
В – координата средней ступени детали-макета $B_м$;
С – координата меньшей ступени детали-макета $C_м$;
5 – величина регулирования суппорта;
7 – координата средней ступени исследуемой детали $B_и$;
8 – координата меньшей ступени исследуемой детали $C_и$;
0 – передаточный коэффициент ПК – получается в процессе расчета.

Перед началом расчета в регистры памяти П7 и П8 следует ввести соответствующие координаты, а в регистр X (индикатор) ввести общую длину исследуемой детали. По сравнению с предыдущей программой, где координаты детали последовательно вводились друг за другом, здесь размеры детали вводятся многократно, а изменяют содержимое регистров ПА, ПВ, ПС в зависимости от того, на какой оснастке необходимо исследовать возможность изготовления данной детали. Если деталь имеет более чем три ступени, программу достаточно просто развить, вставив фрагмент, подобный фрагменту приведенной программы, с 12 по 23 адрес или с 24 по 35. Естественно, что подпрограмма отодвигается, необходимо будет изменить и адрес ссылки на подпрограмму.

Счет по приведенной программе занимает 5–20 с и позволяет объективно судить о пригодности детали к обработке по групповой технологии на имеющихся наладках. Пример анализа деталей, расчета детали-макета и определения пригодности детали для обработки на групповой наладке приведен в табл. 3. Из нее видно, что деталь 5 не прошла при составлении в данную группу по причине слишком малой общей длины, а деталь 7 – из-за длины ступени, не укладывающейся в пределы регулирования суппорта. Те же причины показал микрокалькулятор при исследовании этих деталей для групповой обработки при наладке на деталь-макет. Только при исследовании по второй программе микрокалькулятор конкретизировал ступень, которая препятствует обработке этой детали по групповой технологии: ступень 1 (индикация – 1).

Эскизы деталей, координаты их ступеней и результаты расчета программам 1/34 и 2/34

Таблица

Эскиз	Номер детали								
	1	2	3	4	5	6	7	8	Макет
									
Обозначение ступеней	Координаты ступеней, мм								
A *B* *C*	15 8 3	12 8 4	10 7 3	8 5 2	4 2 1	5 2 1	12 2 1	12 8 7	15,0 8,3 2,8
Индикация при счете по программе 1/34	–	1	2	3	0	4	99	5	–
Время счета, с	–	24	20	17	3	22	11	21	–
Индикация при счете по программе 2/34	100	100	100	100	0	100	–1	100	–
Время счета, с	11	11	11	11	2	11	5	11	–

Деталь для обработки по групповой технологии может иметь отличную от рассмотренной форму, это может быть, например, втулка с наружной проточкой, деталь с проточкой за буртом и др. Принцип подхода остается тем же. Необходимо только помнить, что, проектируя карту наладки на деталь-макет и в общем случае на конкретную деталь, не следует учитывать врезание и перебег инструмента в осевом направлении для автоматов продольного точения, а принимать в качестве расчетных фактические длины ступеней и не учитывать при этом ширину отрезного резца.

3. РАЗМЕРНЫЙ АНАЛИЗ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ

При разработке технологического процесса значительное место занимают различные размерные расчеты, в частности расчеты припусков, размеров заготовок, операционных размеров. На стадии этих расчетов должна быть дана оценка точности и надежности технологического процесса. Размерному анализу технологических процессов в настоящее время отведено большое место [26], поэтому здесь приведен пример автоматизированного расчета размерных цепей на микрокалькуляторе.

При расчете размерных цепей решают два вида задач: проектные и проверочные. Решая проектные задачи, определяют промежуточные операционные размеры, их допуски и размеры заготовки, исходя из окончательных размеров, проведенных на чертеже детали, и выбранного технологом плана обработки этой детали. При проверочной задаче определяют фактические размеры детали и допуски на них либо операционные размеры, припуски и проверяют их соответствие чертежу.

Решая задачу размерного анализа технологических процессов, можно по известным характеристикам составляющих звеньев (размерам) определить характеристики замыкающих звеньев, например, что самое важное в технологическом процессе, наибольший и наименьший на данной технологической операции припуски. Оставив слишком малый припуск, мы рискуем не полностью убрать погрешности формы и установки детали в приспособлении, а чрезмерно увеличив припуск, необоснованно снизим нормативы на механическую обработку, повысим трудоемкость изготовления детали и увеличим непроизводительный расход металла.

Для простоты рассмотрим проверочную задачу.

На рис. 7 приведен эскиз детали, имеющей четыре размера: $A_1 = 50^{+0,2}$ и $A_3 = 19,1^{+0,2}_{-0,4}$, которые выполняют на данной операции; $A_2 = 30 \pm 0,5$, обеспечиваемый вис данной операции (например, этот размер получен на предыдущей операции); A_4 — припуск, являющийся на рассматриваемой операции замыкающим звеном размерной цепи.

Отметим здесь еще два понятия, необходимые для решения задач размерного анализа. Составляющее звено размерной цепи называют уменьшающим, если при его увеличении замыкающее звено уменьшается. В противном случае такое звено называют увеличивающим.

Решая проверочную задачу, необходимо определить, как влияет допуск каждого из составляющих звеньев на значение припуска A_4 и соответственно каковы значения максимального и минимального припусков на данной операции.

При решении проверочной задачи используют следующий порядок вычислений.

1. Определяют координату середины поля допуска каждого составляющего звена

$$\Delta\omega_i = (\Delta_{вi} + \Delta_{нi})/2,$$

где $\Delta_{вi}$ — верхнее предельное отклонение размера; $\Delta_{нi}$ — нижнее предельное отклонение размера.

2. Определяют среднее значение каждого составляющего звена

$$A_{ср\,i} = A_i + \Delta\omega_i,$$

где A_i — номинальное значение размера.

3. Определяют полуколебание каждого составляющего звена

$$\omega_i/2 = (\Delta_{вi} - \Delta_{нi})/2.$$

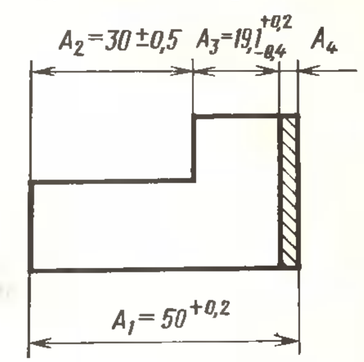


Рис. 7. Эскиз детали к расчету размерных цепей

4. Определяют среднее фактическое значение замыкающего звена

$$A_{\text{ср.ф}} = \sum_{i=1}^{n} \xi_i A_{\text{ср}\,i},$$

где $\xi_i = \pm 1$ – коэффициент, указывающий увеличивающее (+1) или уменьшающее (–1) составляющее звено. В общем случае, когда составляющие звенья не лежат на одной прямой, коэффициент может принимать значения $-1 \leqslant \xi_i \leqslant 1$.

5. Определяют фактическое полуколебание замыкающего звена одним из двух методов:

методом минимума-максимума

$$\omega_{\text{ф}}/2 = \sum_{i=1}^{n} \omega_i/2;$$

вероятностным методом

$$\omega_{\text{ф}}/2 = k\sqrt{\sum_{i=1}^{n} (\omega_i/2)^2},$$

где k – коэффициент, определяющий вероятность выхода регламентированной величины за пределы допуска. Для мелкосерийного производства $k = 1{,}7$; для серийного производства $k = 1{,}2$; для массового производства $k = 1$ [26].

6. Определяют минимальное значение замыкающего звена

$$A_{\min\,\text{ф}} = A_{\text{ср.ф}} - \omega_{\text{ф}}/2.$$

7. Определяют максимальное фактическое значение замыкающего звена

$$A_{\max\,\text{ф}} = A_{\text{ср.ф}} + \omega_{\text{ф}}/2.$$

Необходимо учесть, что при расчете по методу минимума-максимума значения замыкающего звена абсолютны и при соблюдении размеров и допусков составляющих звеньев значение замыкающего звена не может выйти за рассчитанные пределы. В случае вероятностного расчета минимальный и максимальный размеры не носят значения абсолютных, но с некоторым допущением (вероятностью) только попадают в рассчитанный по п.п 6 и 7 диапазон.

На завершающем этапе проверочного расчета размерных цепей сравнивают предельные размеры замыкающего звена с рассчитанным значением припуска (минимальное значение которого определяется погрешностью предыдущей операции, шероховатостью поверхности, погрешностью установки на данной операции и т. д.). Полученный ответ и является результатом и целью расчета.

Алгоритм подробно описан выше, поэтому здесь укажем, что для универсальности программы в ней объединены два метода расчета: минимума-максимума и вероятностный. Условием работы по одному из двух вариантов является значение коэффициента k. Если оно не равно нулю, расчет идет по вероятностной ветви; если равно нулю, то по ветви минимума-максимума. Для исключения ошибок число составляющих звеньев отсчитывается оператором цикла.

Программа 3/34. Размерный анализ технологических размерных цепей:

Сх	П7	П8	1	С/П	П4	2	С/П	П5	3
С/П	П6	4	С/П	4	÷	П3	ИП5	ИП6	+
2	÷	ИП4	+	ИП3	×	ИП7	+	П7	ИП5
ИП6	–	2	÷	П5	ИП2	x=0	43	ИП5	ИП8
+	БП	53	ИП8	ИП2	÷	x^2	ИП5	x^2	+
$\sqrt{x}$	ИП2	×	П8	ИП7	+	П9	ИП7	ИП8	–
П0	ИП4	L1	03	С/П	БП	00			

Содержание регистров памяти:

П1 – счетчик цикла – количество составляющих звеньев размерной цепи;

П2 – к=0; 1; 1,2; 1,7 – коэффициент, учитывающий вероятность выхода расчетной величины за пределы допуска;

П3 – коэффициент, определяющий увеличивающее или уменьшающее звено размерной цепи ξ_i;

П4 – номинальный размер составляющего звена размерной цепи A_i;

П5 – верхнее предельное отклонение составляющего звена $\Delta_{\text{в}i}$;

П6 – нижне предельное отклонение составляющего звена $\Delta_{\text{н}i}$;

П7 – среднее фактическое значение замыкающего звена $A_{\text{ср ф}}$;

П8 – фактическое полуколебание замыкающего звена $\omega_{\text{ф}}/2$;

П9 – максимальное значение замыкающего звена $A_{\max\,\text{ф}}$;

П0 – минимальное значение замыкающего звена $A_{\min\,\text{ф}}$.

Перед началом работы определяют, по какому методу будет производиться расчет, и в зависимости от этого в регистр П2 заносится соответствующее значение k. Определяется число составляющих размерную цепь звеньев, и эта величина заносится в регистр-счетчик цикла П1. Нажимают клавиши В/О и С/П (далее только С/П). Так как во время счета по одной программе в микрокалькулятор вводят сразу три числа и знак четвертого, программа составлена таким образом, что шифр числа, которое необходимо ввести с клавиатуры, индицируется. Индикация 1 означает ввод A_i, индикация 2 – ввод верхнего предельного отклонения, индикация 3 – ввод нижнего предельного отклонения. Если значение предельного отклонения отсутствует, вводят нуль. Индикация 4 означает ввод знака, учитывающего уменьшающее или увеличивающее составляющие звенья. Если $\xi_i = -1$, вводим знак /–/ и нажимаем клавишу С/П; если $\xi_i = 1$, то нажимаем только клавишу С/П.

После отработки по алгоритму оператор цикла вновь возвращает управление для ввода очередного составляющего звена размерной цепи по адресу 03 (индикация 1). Одновременно с этим для удобства пользователя в регистре Y стека сохранено значение номинального размера только что введенного звена. При необходимости его можно посмотреть, нажав клавишу XY. После ввода всех звеньев микрокалькулятор выходит из цикла; при этом номинальное значение последнего введенного звена высвечивается. Это сигнал окончания обработки. В регистрах памяти П7, П8, П9, П0 образуются все расчетные данные.

Для приведенного выше примера результаты расчета в зависимости от принятой величины k приведены в табл. 4.

Таблица

Результаты расчетов размерных цепей при различном коэффициенте k

Обозначение	k		
	0	1,2	1,7
A_4	1,1	1,10	1,1
$\omega_ф/2$	0,9	0,71	1,0
$A_{max\,ф}$	2,0	1,81	2,1
$A_{min\,ф}$	0,2	0,40	0,1

Принимая припуск A_4 в качестве составляющего звена размерной цепи, а в качестве замыкающего технологический размер A_3, можно проводить проектные расчеты.

Аналогично осуществляется расчет размерных цепей при составлении технологических процессов на сборку. Но там в качестве замыкающих звеньев принимают размеры всякого рода компенсаторов и прокладок.

ГЛАВА III.

ЭЛЕМЕНТЫ АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ СРЕДСТВ ТЕХНОЛОГИЧЕСКОГО ОСНАЩЕНИЯ

В последнее время в связи с развитием науки о резании металлов и совершенствовании металлообрабатывающего инструмента возникают все более производительные способы обработки изделий, одновременно повышающие качество обработанной поверхности, точность получаемых деталей и другие показатели надежности. Но такое повышение достигается за счет усложнения процессов и инструмента, применения комбинированного инструмента, сочетания резания с поверхностным пластическим деформированием. Эти процессы не столь универсальны и требуют детальнейшего расчета элементов процесса, например геометрических или кинематических параметров, усилий резания или деформирования и т. д.

Иногда формулы приобретают сложный и неуклюжий вид, тогда разрабатывают графоаналитические или графические методы расчета, но и они, как правило, громоздки и ими на промышленных предприятиях пользуются редко. В лучшем случае предприятие обращается к автору с просьбой рассчитать параметры технологического процесса по предложенным формулам для конкретной детали. Сложность расчета не позволяет использовать новые технологии точно по замыслу автора, а неправильное "на ощупь" их применение должного эффекта не приносит.

Для решения большинства задач подобного рода можно с успехом использовать микрокалькулятор.

Значительная часть расчетных задач связана с решением трансцендентных уравнений. Трансцендентным называется такое уравнение, у которого хотя бы одна из частей не является алгебраической функцией. Вообще трансцендентные уравнения могут быть решены только приближенно.

3. МЕТОДЫ РЕШЕНИЯ ТРАНСЦЕНДЕНТНЫХ УРАВНЕНИЙ

Известно несколько методов решения трансцендентных уравнений [6, 8, 13, 16, 36]. Рассмотрим некоторые из них.

Метод итераций (последовательных приближений). Допустим, что решением уравнения

$$F(x) = 0 \qquad (5)$$

является значение $x = x^*$.

Необходимо найти функцию

$$x = F_1(x) \qquad (6)$$

и начальное приближение решения уравнения (5) x_0.

Подставляя x_0 в правую часть уравнения (6), находим x_1 – первое приближение решения уравнения (5). Полученное значение подставляем в правую часть уравнения (6). Процесс итераций прекращается по условию

$$|x_{k+1} - x_k| \leqslant \delta.$$

Вычислительный алгоритм достаточно ясен, поэтому приведем программу вычисления.

Программа 4/34. Нахождение корня трансцендентного уравнения методом итераций (последовательных приближений):

П1	ПП	18	↑	ИП1	–	x<0	09	/–/	ИП2
–	x<0	15	XY	С/П	XY	БП	00	...	...
...	В/О								

Содержание регистров памяти:

П1 – приближение x_k;

П2 – требуемая точность приближения δ.

Вместо многоточия в программу вписывают подпрограмму вычисления функций (6), вводят в регистр X первое приближение x_0 и нажимают клавиши В/О и С/П. После остановки счета результат является решением уравнения (5) с заданной точностью δ. Интересно отметить, что на адресах 06–08 реализована тремя операторами функция выделения абсолютного значения числа.

Однако данный метод имеет недостаток: область его применения ограничена. Если в окрестности корня производная $|F_1'(x)| < 1$ и начальное приближение достаточно близко к корню, то процесс итерации сходится, в противном случае процесс расходится. Сходимость метода существенна тогда, когда $|F_1'(x)| \ll 1$. Например, уравнение $x^7 + 6x - 5 = 0$ в виде (6) можно представить двояко:

1) $x = \sqrt[7]{5 - 6x}$, производная $x' = 6/(7\sqrt[7]{(5-6x)^6})$;

2) $x = (1/6)(5 - x^7)$, производная $x' = -(7/6)x^6$.

Для первого случая производная меньше единицы при всех $x > 1$ и $x < 2/3$, для второго она существенно меньше единицы при $x < 1$. Поэтому из двух представлений уравнения выберем второе. Корень лежит на отрезке [0; 1], так как очевидно, что при $x = 0$ левая часть уравнения равна –5, а при $x = 1$ левая часть уравнения равна +2. Поскольку на концах отрезка функция имеет различные знаки, график по крайней мере один раз пересекает ось абсцисс, и на данном отрезке имеется корень.

Программа 5/34. Нахождение корней уравнения $x^7 + 6x - 5 = 0$ методом итерации:

П1	ПП	18	↑	ИП1	–	x<0	09	/–/	ИП2
–	x<0	15	XY	С/П	XY	БП	00	ИП1	7
·XY	x^Y	5	XY	–	6	÷	В/О		

При счете по этой программе в регистр П2 введено число $\delta = 10^{-5}$, в регистр X введено число $x_0 = 0{,}3$. В результате расчета получено: $x_k = x^* = 0{,}79875786$. Ориентировочное время расчета составило 75 с.

Метод половинного деления. Метод основан на вычислении значения функции на одном конце отрезка и в середине его. Определяют знаки вычисленной функции для выяснения, лежит ли корень уравнения в этой половине отрезка либо в другой.

Если знаки вычисленной функции разные, корень лежит именно в этой половине отрезка, а если одинаковые, то в другой. Тот отрезок, который имеет различные знаки функции на концах, принимается за новую область расположения решения. Ее снова делят пополам, таким же методом вычисляют значение функции и ее знаки в новой середине отрезка, а на одном конце отрезка это значение определено на предыдущей итерации. Приближение заканчивают тогда, когда длина отрезка станет меньшей, чем заданная погрешность приближения.

Алгоритм метода половинного деления (рис. 8).

1. Ввести значения абсцисс концов заданного отрезка $[A, B]$.
   Перейти к п. 2.
2. Вычислить значение функции на одном конце отрезка $F(A)$.
   Перейти к п. 3.
3. Вычислить абсциссу середины отрезка $C = (A + B)/2$.
   Перейти к п. 4.
4. Вычислить значение функции в середине отрезка $F(C)$.
   Перейти к п. 5.
5. Вычислить произведение значений функции на одном из краев отрезка

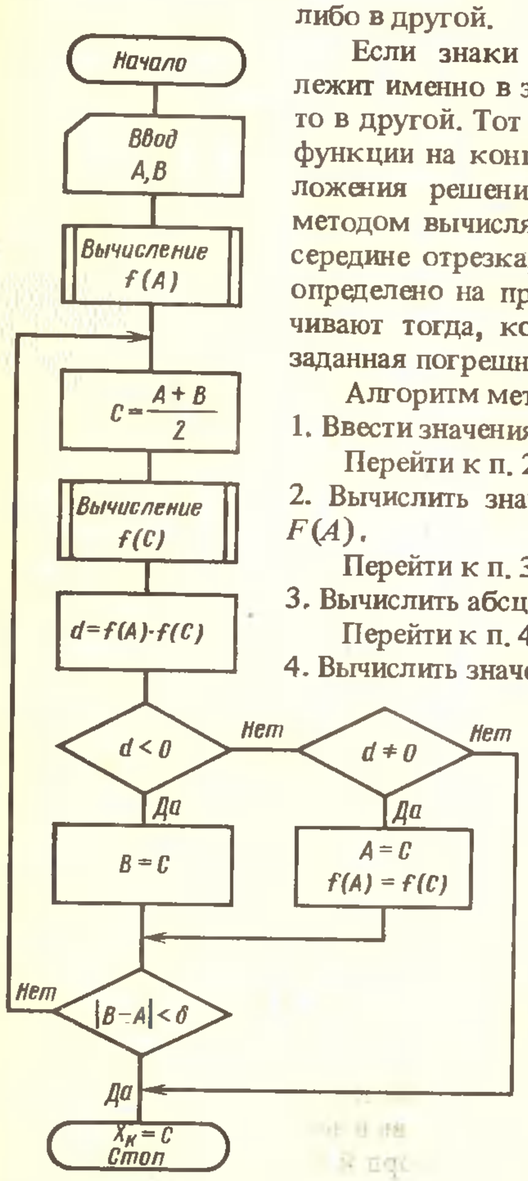


Рис. 8. Блок-схема алгоритма решения трансцендентных уравнений половинным делением

$F(A)$ и в середине его $F(C)$. Если $d = F(A) \cdot F(C)$ меньше нуля, перейти к п. 6; если нет, то к п. 7.

6. Принять $C = B$. Этим шагом отрезок уменьшен с AB до AC, причем новая абсцисса введена в машину снова как B.
   Перейти к п. 9.
7. Если вычисленное в п. 5 произведение равно нулю, индицировать C и перейти к п. 10; если нет, то к п. 8.
8. Принять $A = C$, $F(A) = F(C)$, т. е. локализовать другую половину отрезка AB (уменьшить до BC). Чтобы снова не вычислять значение функции, ее значение в точке C: $F(C)$ присваивается $F(A)$. Перейти к п. 9.
9. Вычислить $|B-A| = \delta_k$. Если $\delta_k < \delta$, индицировать B, перейти к п. 10; если нет, то к п. 3.
10. Конец.

Программа 6/34. Решение уравнений методом половинного деления:

ИПА	ПП	39	П1	ИПА	ИПВ	+	2	÷	ПС
ПП	39	П2	ИП1	×	x<0	30	ИПС	ПВ	ИПА
–	x<0	24	/–/	ИП3	–	x<0	04	ИПВ	С/П
x≠0	28	ИПС	ПА	ИП2	П1	ИПВ	БП	19	...
...	...	В/О							

Содержание регистров памяти:

ПА – абсцисса одного конца отрезка A;
ПВ – абсцисса другого конца отрезка B;
ПС – абсцисса середины отрезка $C = (A + B)/2$;
П1 – значение $F(A)$;
П2 – значение $F(B)$;
П3 – погрешность приближения δ.

Вместо многоточия в программу вписывают подпрограмму вычисления требуемой функции, в регистры ПА и ПВ записывают абсциссы концов отрезка с локализованным корнем, в П3 требуемую погрешность приближения. Нажав клавиши В/О и С/П, ожидают результата. На индикаторе высветится значение решения уравнения с требуемой точностью приближения.

В качестве примера рассмотрим программу решения уравнения $\ln x + x + 2 = 0$ методом половинного деления.

Программа 7/34. Решение трансцендентного уравнения $\ln x + x + 2 = 0$ методом половинного деления. До 39-го адреса записываем программу 6/34:

↑ ln x + 2 + В/О

Начальные значения: $A = 0{,}05$; $B = 100$; $\delta = 10^{-7}$. Среднее время расчета 4 мин; $x_k = 0{,}12002829$.

Метод половинного деления более продолжителен, однако всегда обеспечивает сходимость к корню.

Методы хорд и касательных. В связи с развитием ЭВМ нашли распространение более сложные методы поиска корней, зато они гораздо производительнее двух приведенных выше методов. Эти методы имеют гибкую стратегию, и шаг приближения всякий раз рассчитывается вновь в зависимости от поведения функции. К таким методам относятся методы хорд и касательных [8, 12, 13].

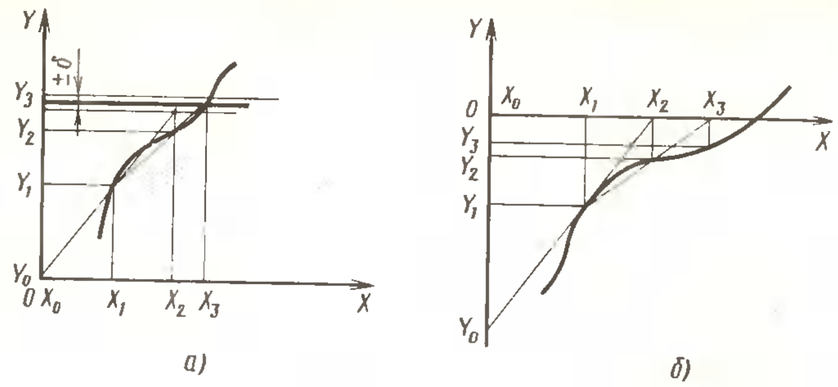


Рис. 9. К решению трансцендентных уравнений методом секущих

Геометрический смысл метода хорд заключается в проведении прямой линии, проходящей через две любые точки монотонного участка графика функции (рис. 9) до пересечения с осью абсцисс, и вычислении очередного приближающего значения функции в точке с абсциссой, равной абсциссе точки пересечения. Вычисленное значение функции является новой точкой, через нее и предыдущую точку проводят новую хорду. Легко доказать, что при малых шагах или больших радиусах кривизны участка графика эта хорда неограниченно приближается к кривой.

Метод касательных (метод Ньютона) аналогичен описанному выше, только при каждой итерации вычисляется производная, по ней определяется угловой коэффициент касательной в данной точке и из точки проводится касательная до пересечения с осью абсцисс. Естественно, что чем более пологая кривая, тем меньше ее первая производная, тем меньше угловой коэффициент касательной, а следовательно, тем больше отстоит точка пересечения касательной и оси абсцисс от последнего вычисленного значения функции.

Для краткости изложения приводим здесь только формулу приращения аргумента для метода хорд [12]:

$$x_{n+1} = -\frac{y_{n-1}}{y_n - y_{n-1}}(x_n - x_{n-1}) + x_{n-1}. \qquad (7)$$

На практике приходится обычно вычислять значение функции не равное нулю, а равное какой либо константе $y_з$. При этом все рассуждения остаются в силе, только на графике функции проводится горизонтальная линия, секущие проводятся до пересечения с ней. При этом формула приближения аргумента запишется следующим образом:

$$x_{n+1} = \frac{y_з - y_{n-1}}{y_n - y_{n-1}}(x_n - x_{n-1}) + x_{n-1}. \qquad (8)$$

Программа 8/34. Решение трансцендентных уравнений методом секущих:

ПП	35	П8	ИПD	–	x<0	08	/–/	ИП6	–
x⩾0	33	ИПD	ИП9	–	ИП1	ИП7	–	×	ИП8
ИП9	–	÷	ИП7	+	ИП1	П7	XY	П1	ИП8
П9	БП	00	ИП1	С/П	...	...	...	...	В/О

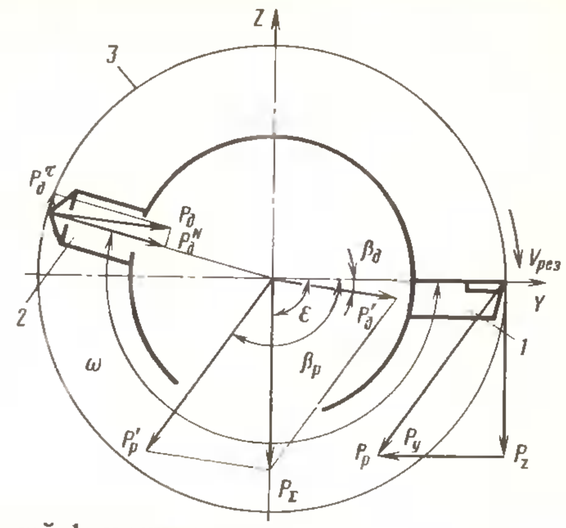


Рис. 10. Расчетная схема комбинированной обработки отверстий:
1 – режущий элемент; *2* – деформирующий элемент; *3* – обрабатываемое отверстие

Содержание регистров памяти:
П0 – заданное значение функции $y_з$ (в том числе может быть нуль);
П1 – текущее значение аргумента x_n;
П6 – заданная погрешность приближения δ;
П7 – регистр предыдущего значения аргумента x_{n-1};
П8 – текущее значение приближаемой функции y_n;
П9 – предыдущее значение приближаемой функции y_{n-1}.

Остальные регистры памяти для приближения не используются, поэтому их можно применять для вычисления значения функции.

2. ПРИЛОЖЕНИЕ ПРОГРАММ РЕШЕНИЯ ТРАНСЦЕНДЕНТНЫХ УРАВНЕНИЙ К ПРОЕКТИРОВАНИЮ ОПТИМАЛЬНЫХ ПАРАМЕТРОВ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ И ОСНАСТКИ

Оптимизация взаимного расположения режущего и деформирующего инструментов при комбинированной обработке. При обработке отверстий комбинированным инструментом, который содержит режущий элемент – резец и деформирующий элемент, например алмазный выглаживатель, можно так расположить деформирующий инструмент относительно режущего, чтобы некоторым образом перераспределить усилия резания и деформирования. Известно, что отжатия режущего инструмента по нормали к обрабатываемой поверхности вносят наибольшие погрешности в обработку. Между тем такие же упругие смещения режущего инструмента по касательной к обрабатываемой поверхности вносят погрешность в обработку на два порядка меньше. Поскольку размер при обработке в наибольшей степени зависит от режущего инструмента – деформирующий инструмент изменяет в основном микрогеометрию и эксплуатационные качества поверхности и свести к нулю результирующую силу обработки комбинированным инструментом не представляется возможным, – предложено так взаимно располагать режущий и деформирующий инструменты, чтобы результирующее усилие обработки было направлено по касательной к обрабатываемой поверхности в точке контакта режущего инструмента [40] (рис. 10).

Предложена формула, позволяющая вычислить взаимное расположение режущего и деформирующего инструментов, обеспечивающая наилучшие режимы обработки отверстий [40]:

$$\omega = \beta_д - \alpha_д + 90^\circ. \qquad (9)$$

Здесь ω – угол взаимного расположения режущего и деформирующего элементов в угловых градусах; угол $\beta_д$ определяется из условия [40]

$$\sin\epsilon = \frac{P_z + P_д \sin\beta_д}{\sqrt{P_р^2 + P_д^2 + 2P_р P_д \cos(\beta_д - \beta_р)}} = \pm 1, \qquad (10)$$

где ϵ, $\beta_р$, $\beta_д$ – соответственно углы между положительным направлением оси Y и равнодействующей P_Σ, усилием резания $P_р$ и усилием деформирования $P_д$, градусы; P_z – тангенциальная составляющая усилия резания, Н;

$\alpha_д$ – угол между тангенциальной составляющей $P_д^T$ и результирующей усилия деформирования $P_д$, выбирается обычно равным 83–88° [19].

Подсчет по формуле (9) не вызывает никаких затруднений. Определение угла $\beta_д$ из условия (10) представляет значительные трудности. Преобразовать выражение (10) таким образом, чтобы изолировать $\beta_д$, слишком сложно и громоздко, тем более трудно будет считать по преобразованной формуле. Гораздо легче решить это уравнение численными методами. А решать это уравнение придется всякий раз, когда мы существенно изменим форму резца, так как при этом изменяется составляющие сил резания, или увеличим припуск на резец или деформирующий элемент.

Все входящие в формулу (10) величины определяются заранее по известным формулам, и для конкретного случая рассчитывается значение $\beta_д$, а затем ω.

Программа 9/34. Программа автоматической оптимизации взаиморасположения режущего и деформирующего инструментов:

ПП	35	П8	1	–	x<0	08	/–/	ИП6	–
x⩾0	33	1	ИП9	–	ИП1	ИП7	–	×	ИП8
ИП9	–	÷	ИП7	+	ИП1	П7	XY	П1	ИП8
П9	БП	00	ИП1	С/П	ИП1	sin x	ИП4	×	ИП5
+	ИП1	ИП2	–	cos x	ИП3	ИП4	×	×	2
×	ИП4	x^2	+	ИП3	x^2	+	$\sqrt{x}$	÷	В/О

Содержание регистров памяти:

П1 – текущее значение угла β_i;
П2 – угол между положительным направлением оси Y и силой резания $P_р$, $\beta_р$;
П3 – сила резания $P_р$, Н;
П4 – сила деформирования $P_д$, Н;
П5 – тангенциальная составляющая силы резания P_z, Н;
П6 – заданная погрешность приближения δ;
П7 – предыдущее значение угла β_{i-1};
П8 – текущее значение приближаемой функции (10) y_i.
П9 – предыдущее значение приближаемой функции y_{i-1}.

Поскольку заданное значение функции равно $y_з = 1$, нет необходимости занимать регистр памяти, и на адресах 03 и 12 программы непосредственно записана 1. В качестве примера рассмотрим задачу оптимизации взаимного расположения режущего и деформирующего элементов. Экспериментальные исследования показали, что при растачивании отверстий ϕ 100 мм; $\beta_р = 106{,}7°$; $P_р = 1044$ Н; $P_z = 1000$ Н. С погрешностью $\delta = 10^{-6}$, начальными условиями $\beta_{д0} = 0$, $\beta_{д1} = 0{,}1$ определить $\beta_д^*$ при $P_д = 400$, 500, 600 и 700 Н. Результаты расчета приведены в табл. 5.

В некоторых случаях график функции асимптотически приближается к горизонтальной прямой, пересечение с которой мы ищем, или имеет рядом с ней локальные экстремумы. При этом важно установить, когда вновь вычисленное значение функции не отличается от предыдущего значения более чем на заданную величину, и вывести программу из цикла итераций. Это можно выполнить заменой оператора вызова заданного значения функции с адресом 03 в программах 8/34 и 9/34 на вызов предыдущего значения функции: ИП9. Но такая задача решается при оптимизационных расчетах, поэтому ниже она будет рассмотрена более подробно.

Таблица 5

Результаты расчета угла взаимного положения режущего и деформирующего элементов

$P_д$, Н	Время расчета, мин	$\beta_д^*$, °	ω^*, °, вычисленное по формуле (9)
400	5,0	41,7	43–49
500	3,5	53,4	55–60
600	3,5	59,9	62–67
700	4,0	64,5	66–72

Расчет параметров затылования металлорежущего инструмента. Для образования режущего клина на металлорежущем инструменте выполняют задний угол. Это угол, образованный плоскостями, касательными к обрабатываемой поверхности и к задней поверхности инструмента и проходящими через вершину инструмента.

Самым простым, но не самым лучшим способом образования заднего угла является заточка инструмента по плоскости. Она позволяет перетачивать инструмент по задней поверхности без использования каких-либо приспособлений. К недостаткам такой заточки следует отнести, во-первых, невозможность переточек фасонного инструмента, во-вторых, значительное ослабление вершины зуба.

По этой причине у тяжелонагруженного и фасонного инструмента заднюю поверхность выполняют по дугам сложных кривых (затылуют), таких, как различные спирали. При образовании на задней поверхности фасонного инструмента площадки износа с нулевым углом имеется возможность переточить инструмент с образованием плоскости на передней поверхности. Профиль при этом остается практически неизменным. Эти два преимущества – неизменность профиля и переточки по плоскости – определили широкое распространение спирали Архимеда.

Но, как и другие методы, способ затылования инструмента по спирали Архимеда не лишен недостков. Важнейшим из них является то, что для получения такой спирали необходимо согласовать два движения: равномерное вращение и возвратно-поступательное движение.

Повысить производительность затылования сложно из-за больших динамических нагрузок при осуществлении быстрого возвратно-поступательного движения. Кроме того, при затыловании нужно применять специальные затыловочные станки и приспособления. Недостатки способа затылования по спирали Архимеда вынуждают технологов искать другие способы. В частности, используют затылование по дуге окружности, центр которой не проходит через центр инструмента (рис. 11).

Этот способ может быть реализован на универсальном заточном станке, однако каждый зуб в этом случае затылуется индивидуально, поэтому производительность его невысока. Кроме того, нет математического аппарата, позволя-

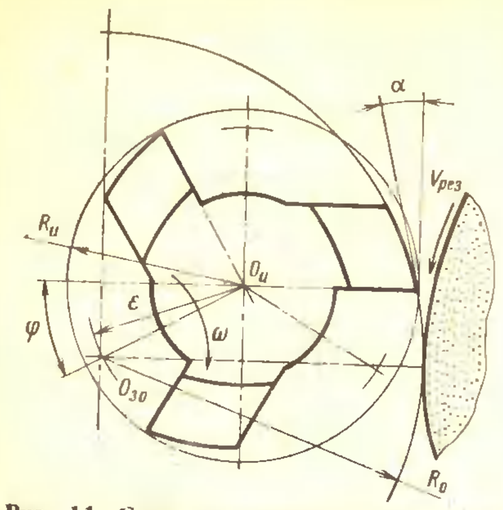


Рис. 11. Схема затылования металлорежущего инструмента по дуге эксцентричной окружности:
$O_и$ – центр окружности инструмента; $O_{з.о}$ – центр затыловочной окружности; ω – формообразующее вращение; $V_{рез}$ – скорость резания; R_0 – радиус затылующей окружности

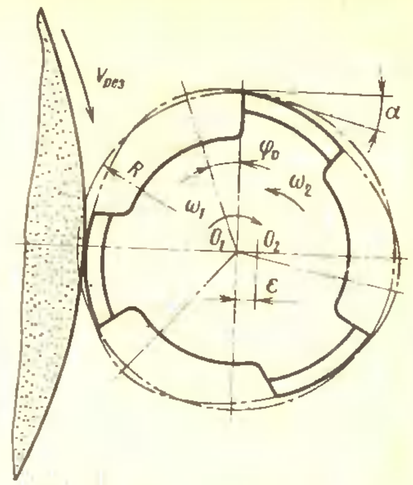


Рис. 12. Схема затылования металлорежущего инструмента по гипотрохоиде

ющего рассчитать требуемый задний угол инструмента в общем случае при таком способе затылования. Предложено затыловать инструмент по гипотрохоиде (укороченной гипоциклоиде), которая кинематически реализуется вращением инструмента вокруг двух параллельных осей с различными по величине и направлению угловыми скоростями [31]. Отношение угловых скоростей должно быть равно числу зубьев затылуемого инструмента (рис. 12).

Разработано приспособление для вращения инструмента вокруг двух параллельных осей O_1 и O_2 с различными угловыми скоростями ω_1 и ω_2 [31]. Однако здесь еще более необходим точный метод расчета углов затылования, так как в отличие от предыдущего способа графически эту задачу решить несравненно сложней.

Рассмотрим, как изменяется задний угол инструмента в различных точках затылованной поверхности для разных кривых затылования. В любой точке поверхности задний угол можно определить как угол между касательной (в плоскости чертежа) к окружности, проходящей через эту точку и имеющей центр в центре вращения инструмента, и касательной к заданной кривой в этой же точке:

$$\alpha = \theta_0 - \theta_к, \tag{11}$$

где θ_0, $\theta_к$ – углы наклона касательных к окружности и к кривой в заданной точке к оси абсцисс.

Известно [13], что угловой коэффициент касательной к любой кривой в заданной точке (тангенс угла наклона ее к оси абсцисс) тождественен производной функции в этой точке:

$$\operatorname{tg}\theta = \frac{dy}{dx} = \frac{dy/d\varphi}{dx/d\varphi}, \tag{12}$$

где φ – полярная координата рассматриваемой точки кривой.

Тогда для касательной к окружности

$$\operatorname{tg}\theta_0 = \frac{r\cos\varphi}{-r\sin\varphi} = -\operatorname{ctg}\varphi.$$

Задний угол, описываемый любой аналитической кривой, можно определить из выражения

$$\alpha = \operatorname{arctg}(-\operatorname{ctg}\varphi) - \operatorname{arctg}\left(\frac{dy/d\varphi}{dx/d\varphi}\right). \tag{13}$$

Рассмотрим некоторые затыловочные кривые.

А. Кривая затылования – гипотрохоида

Используя формулу (13) и параметрическое уравнение гипотрохоиды, запишем выражение для заднего угла инструмента, задняя поверхность которого выполнена по гипотрохоиде [31]:

$$\alpha = \operatorname{arctg}(-\operatorname{ctg}\varphi) - \operatorname{arctg}\left[-\frac{R\cos\varphi - \epsilon(z-1)\cos(z-1)\varphi}{R\sin\varphi + \epsilon(z-1)\sin(z-1)\varphi}\right], \tag{14}$$

где R – максимальный радиус затылуемого инструмента, мм; ϵ – эксцентриситет – расстояние между двумя параллельными осями, вокруг которых вращают инструмент, мм; z – число зубьев затылуемого инструмента.

Очевидно, что выражение (14) трансцендентное и определить параметры затылования φ, ϵ через известные величины α, R, z можно только численными методами. Для этого воспользуемся прикладной программой 8/34.

Коэффициенты уравнения (14) можно нормировать, разделив числитель и знаменатель на ϵ; при этом вместо двух регистров памяти для размещения R и ϵ потребуется один для R/ϵ.

Программа 10/34. Расчет угла установки инструмента при затыловании по гипотрохоиде:

ПП	35	П8	ИПD	–	x<0	08	/–/	ИП6	–
x⩾0	33	ИПD	ИП9	–	ИП1	ИП7	–	×	ИП8
ИП9	–	÷	ИП7	+	ИП1	П7	XY	П1	ИП8
П9	БП	00	ИП1	С/П	ИП1	cos x	ИПА	×	ИП0
ИП1	×	cos x	ИП0	×	–	ИП1	sin x	ИПА	×
ИП0	ИП1	×	sin x	ИП0	×	+	÷	arctg x	ИП1
tg x	1/x	arctg x	XY	–	В/О				

Содержание регистров памяти:

П1 – текущее значение аргумента φ_i, °;
П6 – заданная погрешность приближения δ, °;
П7 – предыдущее значение аргумента φ_{i-1}, °;
П8 – вычисленное значение функции (заднего угла) α_i, °;
П9 – предыдущее значение функции (заднего угла) α_{i-1}, °;
ПD – число на единицу меньшее числа зубьев инструмента $z-1$;
ПА – отношение максимального радиуса инструмента к эксцентриситету R/ϵ;
П0 – заданное значение функции (заднего угла) $\alpha_з$, °.

В качестве примера рассмотрим задачу расчета угла установки инструмента от вершины гипотрохоиды при заданном заднем угле $\alpha_з = 7^\circ$, числе зубьев инструмента $z = 4$, погрешности приближения $\delta = 10^{-6\,\circ}$. Начальные данные $\varphi_0 = 0$ (П7); $\alpha_0 = 0$ (П9); $\varphi_1 = 0{,}1^\circ$ (П1). Отношение радиуса к эксцентриситету $R/\epsilon = 12$ (ПА).

Если два последовательно вычисленных значения функции совпадают с точностью до машинного нуля (при $\delta \leqslant 10^{-6}$), в результате деления на нуль возникает ошибка, которая означает конец расчета.

Время расчета составило 120 с, $\varphi^* = 5{,}544^\circ$ при $\alpha^* = 6{,}999999^\circ$.

Необходимо отметить, что, поскольку погрешность в настоящей задаче непосредственно связана с выходным параметром процесса — заданным задним углом, ее нет смысла выполнять точнее, чем можно измерить этот угол, т. е. точнее, чем $\delta = 0{,}1^\circ$. В предыдущем расчете взята $\delta = 10^{-6}$ для иллюстрации возможностей расчета. Рассчитаем предыдущую задачу, но для $\alpha_з = 35^\circ$ и отношения $R/\epsilon = 5$, $\delta = 0{,}1^\circ$. Время расчета составило 150 с, $\varphi^* = 9{,}4544769^\circ$, $\alpha^* = 34{,}968407^\circ$.

Иногда необходимо при известном угле установки определить требуемое отношение радиуса инструмента к эксцентриситету. В этом случае надо несколько изменить обращение к регистрам памяти, оставив всю программу в целом без изменения.

Программа 11/34. Расчет оптимального отношения радиуса инструмента к эксцентриситету при выполнении задней поверхности инструмента по гипотрохоиде:

ПП	35	П8	ИПD	–	x<0	08	/–/	ИП6	–
x⩾0	33	ИПD	ИП9	–	ИПА	ИП7	–	×	ИП8
ИП9	–	÷	ИП7	+	ИПА	П7	XY	ПА	ИП8
П9	БП	00	ИПА	С/П	...	...	...	...	

Остальная часть программы и распределение регистров памяти такие же, как в программе 10/34.

Исходные данные для расчета: $z = 4$; $\alpha_з = 35^\circ$; $\alpha_0 = 90^\circ$; $R/\epsilon_0 = 0$; $\varphi = 15^\circ$; $\delta = R/\epsilon_1 = 0{,}1$. Время расчета 80 с; $R/\epsilon^* = 5{,}22$.

Б. Кривая затылования — эксцентричная окружность

Используя формулу (13), можно записать для эксцентричной окружности

$$\alpha = \operatorname{arctg}(-\operatorname{ctg}\varphi) - \operatorname{arctg}\left[-\frac{\cos\varphi + \epsilon/R}{\sin\varphi}\right], \qquad (15)$$

где φ — угол поворота радиуса, на котором лежит центр затылуемой окружности относительно оси абсцисс, градусы; ϵ — эксцентриситет, на который сдвинута окружность затылования относительно центра инструмента, мм; R — радиус затылуемого инструмента, мм.

Программа 12/34. Расчет угла поворота оси центра затылуемой окружности относительно оси абсцисс.

ПП	35	П8	ИПD	–	x<0	08	/–/	ИП6	–
x⩾0	33	ИПD	ИП9	–	ИП1	ИП7	–	×	ИП8
ИП9	–	÷	ИП7	+	ИП1	П7	XY	П1	ИП8
П9	БП	00	ИП1	С/П	ИП1	cos x	ИПА	1/x	+
ИП1	sin x	÷	arctg x	ИП1	tg x	1/x	arctg x	–	В/О

Содержание регистров памяти аналогично содержанию регистров памяти в программе 10/34.

В качестве примера рассмотрим задачу затылования инструмента по эксцентричной окружности: $R/\epsilon = 5$; $\alpha_з = 10^\circ$; $\delta = 0{,}1^\circ$; $\varphi_0 = 0^\circ$; $\alpha_0 = 0^\circ$; $\varphi_1 = 0{,}1^\circ$. Время расчета 70 с, $\varphi^* = 69{,}928^\circ$; $\alpha^* = 9{,}970^\circ$.

Поскольку при затыловании по эксцентричной окружности каждый зуб инструмента обрабатывают индивидуально, угол поворота радиуса, на котором лежит центр затыловочной окружности, не может быть больше $360^\circ/(2z)$ с тем, чтобы избежать нежелательного касания необрабатываемого зуба затылующим инструментом.

Для этого случая может быть решена другая задача: при известном угле установки определить отношение радиуса к эксцентриситету для обеспечения заданного угла затылования. Для решения такой задачи необходимо в программе 12/34 первые 35 операторов заменить на операторы программы 11/34.

В качестве примера рассмотрим задачу определения отношения радиуса к эксцентриситету при затыловании по эксцентричной окружности: $\alpha_з = 35^\circ$; $\varphi = 40^\circ$; $\delta = 0{,}1$; $R/\epsilon_0 = \alpha_0 = 0$; $R/\epsilon_1 = 0{,}1$. Время расчета 65 с, $R/\epsilon^* = 0{,}15$. При $\alpha_з = 8^\circ$ и всех данных предыдущей задачи $R/\epsilon^* = 3{,}782$. Время счета около 3 мин.

Поскольку обе рассмотренные кривые затылования при использовании в качестве проектируемого параметра φ имеют экстремальную зависимость $\alpha = F(\varphi)$, может оказаться, что $\alpha_з > \alpha_{max}$; в этом случае микрокалькулятор зациклится или выдаст сообщение об ошибке. Чтобы этого не случилось, желательно пользоваться программами с проектируемым параметром R/ϵ, при которых график функции $\alpha = F(R/\epsilon)$ не имеет экстремума.

Аналогично может быть построена программа для проектирования параметров затылования инструмента с выбором в качестве кривой затылования любой аналитически заданной кривой.

В. Кривая затылования — спираль Архимеда.

При затыловании инструмента по спирали Архимеда нет смысла пользоваться формулой (13), поскольку задний угол можно определить из известной формулы для угла между радиусом кривизны и радиус-вектором спирали Архимеда:

$$\alpha = \operatorname{arctg} 1/\varphi_р = \operatorname{arctg} 180/(\pi\varphi), \qquad (16)$$

где $\varphi_р$ — полярный угол в радианах.

Для изготовления спиралей Архимеда на современных токарно-затыловочных станках необходимо задавать параметр затылования k, который обычно определяют из формулы [44]:

$$\operatorname{tg}\alpha = kz/(2\pi R). \qquad (17)$$

Формула (17) получена из криволинейного треугольника ABC (рис. 13). Однако поскольку распрямить его до треугольника $A'BC$ без растяжения или сжатия сторон невозможно, следовательно, формула (17) приближенная. Для того чтобы определить точное значение угла затылования, найдем его из определения параметра затылования по спирали Архимеда. Параметр k представляет собой уменьшение радиус-вектора спирали при переходе от предыдущего

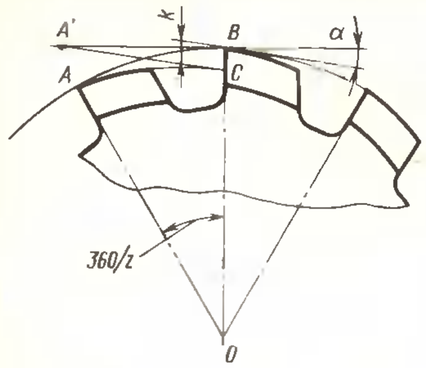


Рис. 13. К определению заднего угла инструмента, затылованного по спирали Архимеда

зуба к последующему. Поэтому для радиус-вектора на одном зубе можно записать

$$R - k = a\varphi,$$

где a — параметр спирали Архимеда, а для радиус-вектора на соседнем зубе

$$R = a(\varphi + 360/z).$$

Так как параметр a входит в оба уравнения, то разделив второе уравнение на первое, его можно исключить:

$$R/(R - k) = 1 + 360/(z\varphi).$$

Находя из последующего уравнения φ и упрощая, получим

$$\varphi = \frac{360}{z}\left(\frac{R}{k} - 1\right).$$

Подставив последнее уравнение в уравнение (16) и учитывая, что φ выражено в градусах, получим

$$\operatorname{tg}\varphi = \frac{zk}{2\pi(R-k)}. \qquad (18)$$

Сравнивая формулы (17) и (18), легко заметить разницу между значениями заднего угла, полученными по точной (18) и приближенной (17) формулам.

Для вычисления параметра k при затыловании по спирали Архимеда нет необходимости прибегать к численным методам. Преобразуя формулу (18), находим

$$k = \frac{2\pi R \operatorname{tg}\alpha}{z + 2\pi \operatorname{tg}\alpha}. \qquad (19)$$

Расчет параметров установки инструмента при обработке резьбы ротационными резьбовыми резцами. Для повышения производительности и точности при нарезании многозаходных резьб на универсальных токарных станках предложен способ без использования делительных устройств (А.с. 1134319 СССР, МКИ³ B23G1/00).

Способ заключается в том, что ротационный резьбовой резец, выполненный в виде обычного косозубого зубчатого колеса, устанавливают ниже центра заготовки, чтобы обеспечить кинематический задний угол, а следовательно, резание, и ось резца поворачивают на угол от вертикали в направлении осевой подачи (рис. 14,*а*, *б*). Наклонное расположение резца заставляет кинематический задний угол изменяться от максимального в зоне врезания до нулевого в зоне выхода режущей кромки из зоны резания. Поскольку на выходе из зоны резания кинематический задний угол равен нулю, то происходит, во-первых, беззазорное зацепление косозубого колеса с ранее обработанным участком вин-

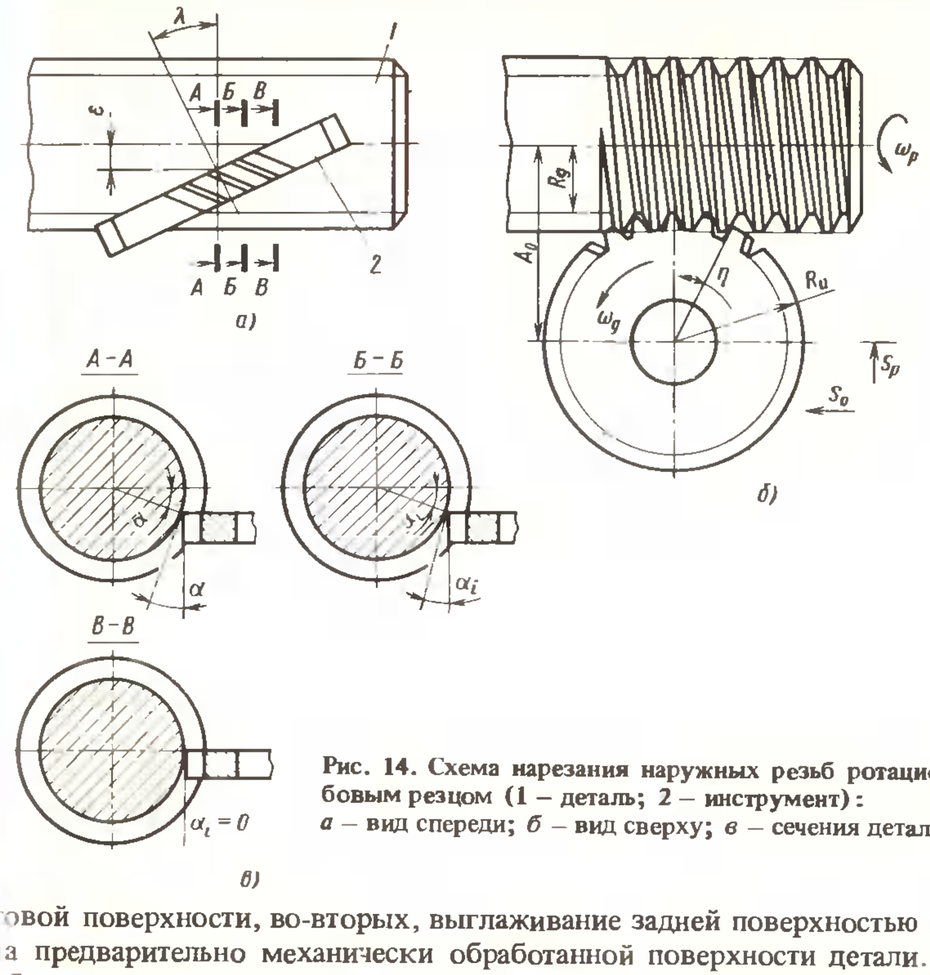


Рис. 14. Схема нарезания наружных резьб ротационным резьбовым резцом (1 – деталь; 2 – инструмент):
а – вид спереди; *б* – вид сверху; *в* – сечения детали

товой поверхности, во-вторых, выглаживание задней поверхностью инструмента предварительно механически обработанной поверхности детали. Последнее обстоятельство положительно сказывается на режуще-деформирующем инструменте, так как изнашивание инструмента происходит параллельно задней поверхности. Здесь можно говорить о размерном изнашивании, но не затуплении инструмента, вследствие того, что статический задний угол инструмента как при заточке, так и в процессе резания останется равным нулю. Технологический процесс изготовления инструмента в виде шлифованного косозубого колеса существенно проще, чем технология изготовления косозубых долбяков для подобных работ.

Однако расчеты, которые необходимо выполнить, несколько сложнее, чем при нарезании резьбы на зубо- или резьбофрезерных станках. Рассмотрим основные условия, необходимые при обработке резьбы по предлагаемому способу.

Координату установки резца ниже центра определяют из формулы

$$\epsilon = R_д \sin\alpha,$$

где $R_д$ — внутренний радиус резьбы детали, мм; α — кинематический задний угол в диаметральной плоскости инструмента (при нарезании резьбы его обычно принимают равным 3–6°).

Натяг при повороте радиус-вектора инструмента от диаметральной плос-

кости в сторону выхода инструмента из зоны резания можно выразить следующим образом:

$$\Delta = A_i - A_0, \tag{20}$$

где $A_i = R_и \cos\eta + R_д \cos\alpha_i$ – теоретическое расстояние, которое получается при повороте радиус-векторов инструмента и заготовки и смещении точки касания от сечения *Б–Б* к сечению *В–В* (рис. 14,*в*); $R_и$ – радиус инструмента; η – угол поворота радиус-вектора инструмента; $A_0 = R_и + R_д \cos\alpha$ – расстояние между осями детали и инструмента в проекции на основную плоскость.

Кинематический задний угол при повороте радиус-векторов инструмента и заготовки можно определить из формулы

$$\alpha_i = \arcsin\,[\sin\alpha - (R_и/R_д)\sin\lambda\sin\eta], \tag{21}$$

где λ – угол отклонения оси инструмента от вертикали.

Для выполнения условий предложенного способа нарезания резьбы необходимо, чтобы при нулевом кинематическом заднем угле $\alpha_i = 0$ сохранился натяг в зоне обработки.

Запишем условие образования нулевого кинематического заднего угла:

так как $\alpha_i = 0$, то $\arcsin\alpha_i = 0$, следовательно, из формулы (21) можно записать

$$\sin\alpha = (R_и/R_д)\sin\lambda\sin\eta,$$

откуда

$$\sin\eta = (R_д/R_и)(\sin\alpha/\sin\lambda). \tag{22}$$

Задаваясь значениями $R_и$, $R_д$, λ и α, определим η по формуле (22). Подставив η в формулу (20), вычислим натяг и сравним его с заданным. Если их значения не совпадают, корректируем один из параметров формулы (22) по формуле (8) и вновь вычисляем новое значение натяга. Вычисление прекращаем, когда разница в значениях заданного и расчетного натягов станет меньше наперед заданной величины δ.

Программа 13/34. Расчет параметров инструмента при обработке резьбы ротационными резьбовыми резцами:

ПП	35	П8	ИПD	–	x<0	08	/–/	ИП6	–
x⩾0	33	ИПD	ИП9	–	ИП1	ИП7	–	×	ИП8
ИП9	–	÷	ИП7	+	ИП1	П7	XY	П1	ИП8
П9	БП	00	ИП1	С/П	1	ИП4	sin x	ИП1	sin x
÷	ИП3	ИП2	÷	×	x^2	–	$\sqrt{x}$	ИП2	×
ИП3	+	ИПА	XY	–	В/О				

Содержание регистров памяти:

П1, П7 – угол отклонения оси инструментов от вертикали λ_i, λ_{i-1};

П2 – радиус инструмента $R_и$;

П3 – радиус детали $R_д$;

П4 – кинематический задний угол α;

П6 – заданная точность приближения δ;

П8, П9 – расчетный натяг Δ_i и Δ_{i-1}, соответственно;

ПА – межосевое расстояние A_0;

П – заданный натяг $\Delta_з$.

Операторами 35, 45, 46, 47 вычисляем косинус по формуле $\cos x = \sqrt{1 - \sin^2 x}$, т. е. эти операторы выполняются существенно быстрее, чем $\arcsin x$ и $\cos x$.

Исходные данные $R_д = 14$ мм, $R_и = 50$ мм, $\alpha = 4°$, $\Delta_з = 0{,}005$ мм, $\lambda_0 = 10°$ (П7), $\lambda_1 = 15°$ (П1), $\Delta_0 = 0$ (П9), $\delta = 10^{-5}$.

Время счета составило 100 с. В результате получено $\lambda^* = 29{,}5°$. Для $\Delta_з = 0$ и тех же исходных данных получено $\lambda^* = 32°$. Время расчета составило 120 с.

Если необходимо рассчитать диаметр инструмента при заданных остальных параметрах, программу следует изменить.

Для этого в регистр П5 записать угол λ (изменив обращение в подпрограмме ИП1 на ИП5), а радиус инструмента занести в П1 (соответственно изменив обращение в подпрограмме ИП2 на ИП1). Кроме того межосевое расстояние необходимо каждый раз вычислять в подпрограмме, но не обращаться к регистру памяти ПА.

Рассмотрим задачу расчета радиуса инструмента при остальных известных параметрах: диаметр детали 28 мм ($R_д = 14$); $\lambda = 25°$: $\alpha = 3°$; $\delta = 10^{-2}$; $R_{и0} = 30$ мм; $R_{и1} = 31$ мм; $\Delta_з = 3$ мкм $= 0{,}003$ мм; $\Delta_0 = 0$. Время расчета 70 с, $R_и^* = 64{,}4$ мм.

В заключение отметим, что пользование прикладными программами требует внимательного отношения к исследуемой функции, в особенности к определению начальных значений, шага и экстремального значения исследуемой функции, если таковое имеется. Так что прикладные программы вовсе не отрицают приближенного аналитического исследования функции, а, наоборот, требуют его. При этом применение программ станет более эффективным.

Приведенные примеры не исчерпывают всех возможностей составления и решения трансцендентных уравнений в технологии машиностроения. Они иллюстрируют подход к решению и дают примеры применения прикладных программ.

3. ОПТИМИЗАЦИОННЫЕ РАСЧЕТЫ В ТЕХНОЛОГИИ МАШИНОСТРОЕНИЯ

При расчете режимов резания и определении трудоемкости изготовления и сборки деталей, в особенности в крупносерийном и массовом производстве, решающее значение имеет выбор варьируемых параметров технологических режимов обработки.

В некоторых случаях варьируемые параметры так входят в функцию производительности обработки, что образуют локальные или глобальные экстремумы. Например, при назначении большого периода стойкости инструмента получаем малые значения расчетной скорости резания, но при этом весьма невелики затраты на переустановку, переточки и подналадки инструмента. При уменьшении периода стойкости расчетное значение скорости резания возрастает, снижается машинное время обработки, однако растет вспомогательное. Поэтому при резании металлов зависимость производительности обработки от периода стойкости инструмента образует глобальный экстремум.

Здесь очевидна задача оптимизации, достаточно хорошо разработанная математиками и экономистами. Суть задачи сводится к поиску целевой функции

$$F(x_i) = A_{\min}, \tag{23}$$

где x_i – независимые переменные $1 \leqslant i \leqslant n$ при k ограничениях $(k \leqslant 2n)$

$$(x_k) \sim C_k. \tag{24}$$

Известен алгоритм последовательного поиска минимума функций многих переменных [38], применяемый для программируемых микрокалькуляторов, при котором независимой переменной x_i придают приращение с постоянным шагом Δx_i до тех пор, пока вновь вычисленное значение функции не станет больше предыдущего:

$$F(x_i) > F(x_{i-1}). \tag{25}$$

Вычисление продолжают с шагом

$$\Delta x_{i+1} = -\Delta x_i/a. \tag{26}$$

Для окончания вычислений используют условие $|\Delta x_i| \leqslant \delta$.

После достижения условия (25) переменную закрепляют и переходят к минимизации функции по следующей переменной. Изменение шага (26) проводят после нахождения минимума с шагом Δx_i для последней независимой переменной. Такая стратегия называется последовательным координатным спуском.

На программируемых микрокалькуляторах удобно организовать оптимизацию с помощью оператора косвенного обращения к памяти, поскольку алгоритм приближения аналогичен для всех переменных. Одновременно с этим изменение номера оптимизируемой переменной можно осуществить оператором цикла; при этом на выходе из цикла необходимо изменить шаг по формуле (26). Косвенное обращение внутри цикла к регистрам памяти с содержанием оптимизируемых переменных можно организовать оператором КИП ↑ (КП ↑). Этот оператор аналогичен оператору КИП0 (КП0), но не изменяет содержимого регистра П0 [39].

Программа 14/34. Минимизация функций многих переменных (до девяти) методом последовательного координатного спуска:

ИПВ	КИП ↑	+	КП ↑	ПП	26	ИПС	XY	ПС	–
x < 0	00	L0	00	· №	П0	ИПВ	ИПА	÷	ПВ
x^2	ИПD	–	x < 0	00	С/П	...	...	...	В/О

Содержание регистров памяти:

П1, П2, ..., П9 – независимые переменные $x_{1i}, x_{2i}, \ldots, x_{9i}$;

П0 – число независимых переменных;

ПА – значение делителя шага a со знаком минус $(-a)$;

ПВ – начальный шаг оптимизации Δx_0;

ПС – вычисленное значение функции $F(x_i)$, первоначально в регистр ПС заносят большое число (например, 10^{50});

ПD – квадрат погрешности приближения δ^2.

В предложенной программе условие $|\Delta x_i| \leqslant \delta$ заменено эквивалентным $\Delta x_i^2 \leqslant \delta^2$, что позволяет сэкономить оператор в программной памяти по сравнению с выделением абсолютного значения, реализованного в программе 5/34.

Вместо знака № в программу записывают число оптимизируемых переменных, вместо многоточия – программу нахождения значения исследуемой функции. Заканчиваться программа должна так, чтобы в операционном регистре X было вычисленное значение функции. Программа занимает 27 операторов программной памяти, но ее можно сократить на три оператора, заменяя отсылку к подпрограмме непосредственным вычислением функции после оператора с адресом 03 и отодвинув остальную часть программы приближения.

Важную роль при поиске минимума функции методом последовательного спуска имеет выбор делителя шага a. Его рекомендуют выбирать больше единицы [37, 38, 39], но оптимальное значение не определено.

Рассчитаем оптимальное значение делителя шага при следующих допущениях:

1) минимум функции определен с точностью до начального шага;
2) функция симметрична в пределах шага относительно минимума;
3) точки, в которых происходит изменение величины и знака шага, имеют равновероятное распределение в пределах одного шага;
4) минимизируется функция одной переменной.

Определим максимальное и минимальное число шагов, необходимых для прохождения в одном направлении. Предположим, что на $(i + 1)$-м шаге (рис. 15) получилось $(x_{i+1} - x^*) < (x^* - x_i)$. Тогда по условию 2: $F_{i+1} < F_i$, и в соответствии с алгоритмом необходимо сделать шаг в направлении $(x_i - x_{i+1})$ вправо. Значение F_{i+2} превысит F_{i+1}, и алгоритм потребует изменения шага. Для возврата к экстремуму с уменьшенным в a раз шагом необходимо сделать a шагов до точки x_{i+1} и затем $(a/2 + 1)$ шаг для перехода за экстремум и нового изменения шага. Следовательно, можно определить максимальное число шагов

$$n_{\max} = 1{,}5a + 1. \tag{27}$$

Сдвиг отрезка $[x_i; x_{i+1}]$ в направлении x_{i+1} относительно экстремума x^* приведет к тому, что $(x_{i+1} - x^*) > (x^* - x_i)$, а следовательно $(F_{i+1} > F_i)$, и здесь шаг будет немедленно изменен. Для этого случая минимальное число шагов составит

$$n_{\min} = 0{,}5a + 1. \tag{28}$$

Очевидно, всякое иное расположение отрезка $[x_i; x_{i+1}]$ относительно экстремума потребует промежуточного между $n_{\max}$ и $n_{\min}$ числа шагов.

Используя допущение 3, определим математическое ожидание числа шагов за один цикл итераций:

$$\bar{n} = (n_{\max} + n_{\min})/2 = a + 1. \tag{29}$$

Число циклов можно определить из условия $|\Delta x_0/a^m| < \epsilon$. Учитывая, что микрокалькулятор выполняет целое число делений шага,

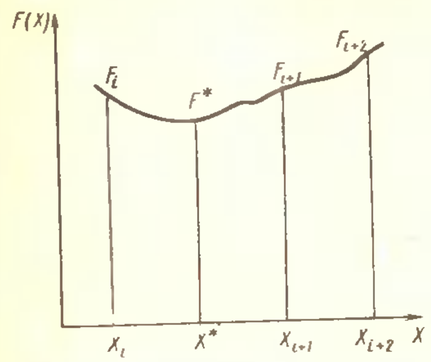
$$m = \mathrm{E}\left\{\log_a \frac{\Delta x_0}{\epsilon}\right\} = \mathrm{E}\left\{\frac{\lg(\Delta x_0/\epsilon)}{\lg a}\right\}, \tag{30}$$

где E $\{\}$ – целая часть числа.

Тогда общее число шагов до достижения заданной точности

$$N = \bar{n}\, m. \tag{31}$$

Поскольку функция (31) имеет разрывы и недифференцируема, приведем график зависимости общего числа шагов до достижения заданной точности от



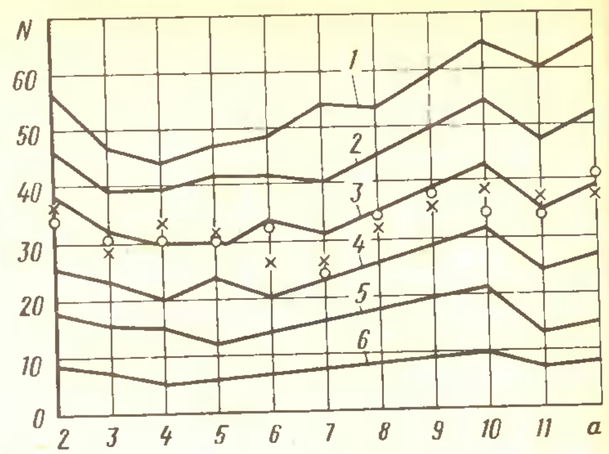


Рис. 15. К определению числа итераций при минимизации функции на одном цикле приближения

Рис. 16. Графики зависимости математического ожидания числа итераций при минимизации методом последовательного координатного спуска:
1 – $\epsilon/\Delta x_0 = 10^{-6}$; *2* – $\epsilon/\Delta x_0 = 10^{-5}$; *3* – $\epsilon/\Delta x_0 = 10^{-4}$; *4* – $\epsilon/\Delta x_0 = 10^{-3}$; *5* – $\epsilon/\Delta x_0 = 10^{-2}$; *6* – $\epsilon/\Delta x_0 = 10^{-1}$ (фактическое число шагов для функции (32) при $\epsilon/\Delta x_0 = 10^{-4}$ обозначено X – для $n=5$, ○ – для $n=7$)

значения делителя шага a (рис. 16). Из графика видно, что значение делителя a, при котором общее число шагов минимально, в общем случае можно принять равным четырем, в частности, для конкретной точности выбрать из графика (рис. 16).

Необходимо помнить, что математическое ожидание носит вероятностный характер, зависит от исходной точки приближения, поведения графика функции вблизи экстремума, поэтому в каждом конкретном случае может несколько отличаться от расчетного.

При экономических и технических расчетах часто в качестве независимых переменных выступают физические параметры, которые не могут быть отрицательными. В этом случае в программу следует ввести проверку неотрицательности по всем переменным.

Программа 15/34. Оптимизация функций многих переменных с условием неотрицательности переменных:

ИПВ	КИП↑	+	x⩾0	14	КП↑	ПП	28	ИПС	XY
ПС	–	x<0	00	L0	00	№	П0	ИПВ	ИПА
÷	ПВ	x^2	ИПD	–	x<0	00	С/П	...	В/О

В ряде задач науки и техники возникает необходимость работать с переменными, значения которых отличаются на два или даже на четыре порядка. Например, в теории резания и технологии машиностроения необходимо оптимизировать скорость резания и стоимость обработки по стойкости инструмента (сотни минут), глубине резания (единицы миллиметров), подаче (сотые доли миллиметра). Для таких задач приведенные программы неприменимы из-за несравнимости значения шага для всех переменных. В подобных задачах можно работать с относительным шагом оптимизации, а абсолютное значение шага получать умножением относительного шага на значение независимой переменной.

Программа 16/34. Оптимизация функций многих переменных с изменяемым шагом оптимизации при переходе к каждой переменной:

КИП↑	ИПВ	X	П9	КИП↑	ИП9	+	КП↑	ПП	30
ИПС	XY	ПС	–	x<0	04	L0	00	№	П0
ИПВ	ИПА	÷	ПВ	x^2	ИПD	–	x<0	00	С/П
...	...	...	В/О						

В некоторых случаях значение абсолютного шага оптимизации, полученное по адресу 02 программы, можно хранить в стеке, для этого вместо фрагмента программы 16/34: П9 КИП↑ ИП9 + следует записать: ↑ F↻ КИП↑ +, а подпрограмму начинать оператором XY и завершать так, чтобы в стековых регистрах Y и Z было значение абсолютного шага оптимизации. Это несколько усложнит программу, зато сбережет один регистр памяти.

При оптимизации с переменным шагом важную роль играет выбор направления поиска минимума и начальных точек x_{0i}.

Если $x_i \in [0;\ \infty]$, то при выборе относительного шага оптимизации $\Delta_0 = 1$ при любом $x_0 > x^*$ и направлении оптимизации к нулю всегда попадаем в точку $x_1 = 0$. Это требует немедленного изменения шага, начиная с исходной точки, поскольку при $x_i = 0$ функция не определена. Поэтому следует начинать минимизацию с шага $\Delta_1 = \Delta_0/a$.

Допустим далее, что $x_0 = lx^*$, где l – некоторое целое число. С учетом сказанного начальный относительный шаг оптимизации принят $\Delta_0 = 1/a$. Значение абсолютного шага оптимизации будет превышать x^* в l раз, следовательно среднее число шагов на первом цикле итераций составит

$$\bar{n} = la + 1$$

и приведет к увеличению общего числа шагов оптимизации.

В случае, если начальная точка приближения находится ближе к нулю, чем экстремум, и шаг приращения выбран в его направлении (положительное направление), то при условии $x_0 = (1/l)x^*$ и относительном шаге оптимизации $\Delta_0 = 1$ экстремум будет достигнут за l итераций.

Последующее уменьшение относительного шага оптимизации в a раз вызовет уменьшение абсолютного шага оптимизации в a/l раз. При этом экстремум будет достигнут с точностью абсолютного шага. На второй итерации число шагов до достижения экстремума составит $\bar{n} = a/l + 1$.

На дальнейших циклах оптимизации математическое ожидание числа шагов на каждой итерации описывается формулой (31).

Из этого следует вывод, что при оптимизации с относительным шагом желательно выбирать такие начальные значения, при которых $x_{0i} \approx \Delta x_{0i}$, а начальный относительный шаг оптимизации принять равным $\Delta_0 = 1$. Оптимальной стратегии поиска минимума, начиная с бо́льших значений переменных и относительным шагом, направленным в отрицательном направлении, при последовательном координатном спуске не существует.

Стратегию поиска можно улучшить, применив направленный поиск. Суть его заключается в том, что после $F_{i+1} > F_i$ делают шаг назад и только после

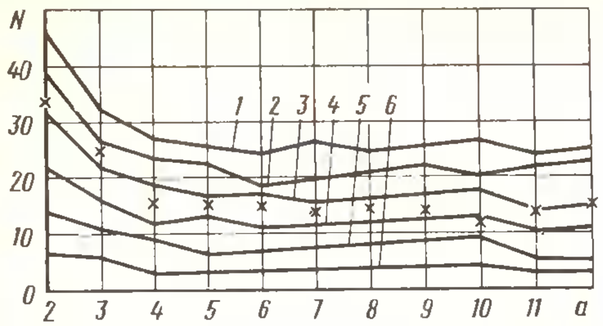


Рис. 17. Графики зависимости математического ожидания числа итераций при минимизации функции методом улучшенного последовательного координатного спуска (обозначения те же, что на рис. 16)

этого кратно уменьшают шаг с изменением его знака. После этого выполняется пробный шаг в новом направлении, и в случае успеха этот шаг закрепляется. При увеличении функции знак изменяют. Такая стратегия позволяет получить меньшее математическое ожидание числа шагов на одном цикле: $\bar{n} = a/4 + 2$ (при $n_{max} = a/2 + 2$ и $n_{min} = 2$). Это приводит к уменьшению общего числа шагов (31) (рис. 17).

Программа 17/34. Минимизация функций многих переменных методом направленного координатного спуска:

ПП	22	ПВ	ПП	22	ПП	34	ПС	L0	00
№	П0	ИПВ	ИПА	÷	ПВ	x^2	ИПD	–	x < 0
00	С/П	ПП	33	ИПС	XY	ПС	–	x < 0	03
ИПВ	/–/	В/О	ИПВ	КИП↑	+	КП↑	...	...	В/О

Важным свойством предлагаемого алгоритма является устойчивость решения. Если на каком-либо шаге поиск приводит в точку экстремума, на всех последующих циклах будет выполнено только по два шага: в сторону от экстремума и к нему. Из этого свойства вытекают два важных для практики следствия.

1. Если на каком-либо шаге получен экстремум, дальнейшее уменьшение шага не приводит к повышению точности вычислений (из-за погрешности округления в микрокалькуляторе точность может даже ухудшаться).

2. Значение делителя a должно быть достаточным, чтобы на первых циклах итераций достичь экстремума (см. рис. 17). Из графика видно, что изменение значения делителя шага от 6 и выше не оказывает существенного влияния на число шагов оптимизации. Удобным оказывается делитель шага $a = 10$, вследствие простоты его программного ввода без использования регистров памяти и быстрого попадания в экстремум в случае целочисленных оптимальных значений целевой функции и целочисленных начальных условий приближения.

Можно программно реализовать следствие 1.

Для этого нужно организовать проверку значения переменной на входе в цикл приближения и при выходе из цикла. Если эти значения равны, вычисление прекратить, если нет, то продолжить.

Программа 18/34. Ускоренная минимизация функций многих переменных методом направленного координатного спуска:

ПП	27	ПВ	ПП	29	ПП	41	ПС	КИП↑	ИП9
–	x ≠ 0	26	L0	00	№	П0	ИПВ	ИПА	÷
ПВ	x^2	ИПD	–	x < 0	00	С/П	КИП↑	П9	ПП
40	ИПС	XY	ПС	–	x < 0	03	ИПВ	/–/	В/О
ИПВ	КИП↑	+	КП↑	...	...	...	...	В/О	

Можно существенно уменьшить время вычислений, если воспользоваться регистром предыдущего результата и не рассчитывать значения функции по программе в случае неудачной попытки при шаге назад, а просто присвоить соответствующим регистрам памяти те значения, которые были в них до данной попытки. Это регистры оптимизируемой переменной и значения функции.

Программа 19/34. Ускоренная оптимизация функций многих переменных направленным координатным спуском без вычисления значения функции на обратных шагах после неудачных попыток с относительным шагом оптимизации и условием $X > 0$:

ПП	52	ПС	КИП↑	ИПВ	×	ПА	ПП	27	ПА
ИПА	ПП	27	L0	03	№	П0	ИПВ	ВП	1
/–/	ПВ	ИПD	–	x < 0	03	С/П	КИП↑	+	КП↑
x ⩾ 0	45	x ≠ 0	45	ПП	52	ИПС	XY	ПС	–
x < 0	10	вх	+	ПС	КИП↑	ИПА	/–/	+	КП↑
вх	В/О	...	...	...	...	В/О			

Первые три оператора необходимы для вычисления значения функции при начальных значениях оптимизируемых переменных. С помощью операторов с адресами 03–06 вычисляется абсолютное значение шага оптимизации по каждой переменной. Подпрограмма с адреса 27 выполняет приращение оптимизируемой переменной, проверку условия $x > 0$. При выполнении этого условия вычисляется значение функции и сравнивается с предыдущим вычисленным значением. Если $F_i < F_{i-1}$, то управление передается на адрес 10 и производится новый шаг итерации. Если условие $F_i < F_{i-1}$ не выполняется, значит, попытка неудачная, в регистре ПС восстанавливается предыдущий результат (меньшего значения функции – операторы с адресами 42–44), в регистре с минимизируемой переменной – также предыдущий результат (адреса 45–49), в операционный регистр X вызывается отрицательное значение абсолютного шага оптимизации, и при выходе из подпрограммы на адрес 09 это отрицательное значение шага закрепляется. Если условие $x_i > 0$ не выполняется, в регистре с оптимизируемой переменной восстанавливается предыдущий результат (адреса 45–49), и управление передается на выход из подпрограммы.

В связи с тем, что переменная не может быть равной или меньше нуля, а калькулятор осуществляет переход только при невыполнении условия, по адресам 30–34 осуществляется проверка $x_i > 0$.

Поскольку в программе отрицательным может быть только значение абсолютного шага оптимизации, а относительный шаг оптимизации – величина всегда положительная, нет необходимости выделять абсолютное значение. Относительный шаг оптимизации сравнивается непосредственно с заданной погрешностью вычислений. Блок-схема алгоритма, представленного программой 18/34, показана на рис. 18.

Эта программа позволяет работать с абсолютным шагом оптимизации, для этого по адресу 03 записывают 1 вместо КИП↑.

Для проверки допущений 2 и 3 использовалась функция [36]

$$F(x_i) = \exp\left(\sum_{i=1}^{n} x_i\right) / \prod_{i=1}^{n} x_i^i \qquad (32)$$

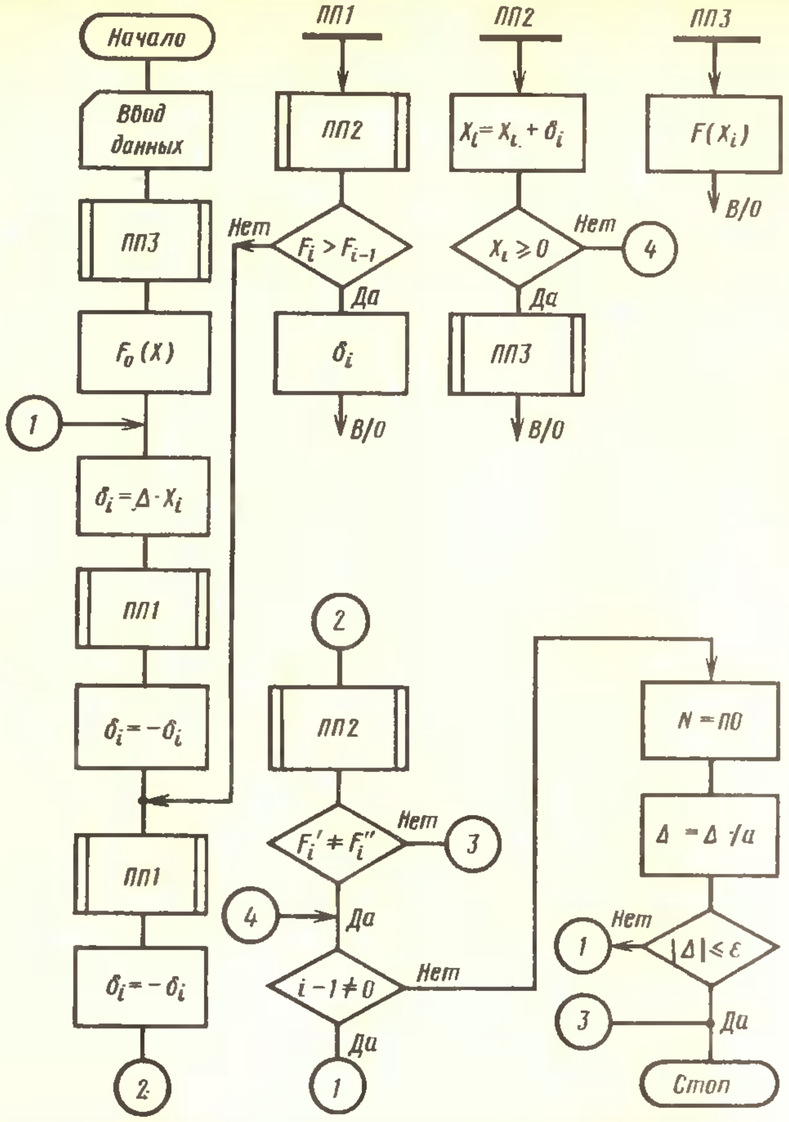


Рис. 18. Блок-схема алгоритма минимизации функции многих переменных с ограничениями x > 0 методом улучшенного направленного координатного спуска

при $n = 5$ с минимумом при $x_i = i$. Начальные условия задавались различными. Начальный шаг минимизации $\Delta x_0 = 1$. Функция минимизировалась по различным программам 14/34 – 19/34. Чтобы обеспечить допущение 1, было просчитано число шагов до первого изменения шага. Для $n = 5$ оно составило 20. Учет числа шагов осуществлял микрокалькулятор операторами ИП9 1÷П9. Возведение в степень, как очень медленная функция, было заменено последовательным возведением в квадрат и умножением [37]. Среднее значение числа шагов до получения заданной точности определяли делением общего числа шагов N на количество переменных (см. рис. 16).

Алгоритм направленного последовательного спуска проверяли при $n = 5$. Для усреднения результатов в качестве начальных условий каждый раз брали

0,1; 0,2; 0,3; 0,4; 0,5. На рис. 17 показано среднее значение числа ходов на одну переменную в зависимости от значения делителя для $n = 5$; $\Delta x_0/\delta = 10^4$. Спад кривой при $a = 10$ обусловлен достаточно быстрым попаданием переменных в экстремум. С учетом теоретической и экспериментальной кривых можно рекомендовать принимать делитель в этом алгоритме равным 10.

Алгоритм ускоренной минимизации также проверяли на функции (32). Для исключения быстрого попадания в экстремум начальные условия принимали равными 0,1; 0,2; ...; 0,5; 0,555. Сравнительная эффективность алгоритмов минимизации в зависимости от начальных условий приведена в табл. 6.

Некоторое расхождение теоретических и экспериментальных данных обусловлено, по-видимому, во-первых, несимметричностью графика функции около экстремума, во-вторых, погрешностью вычислений в микрокалькуляторе, так как оптимальные значения переменных отличаются от точных расчетных на разную величину, и, наконец, в-третьих, небольшим числом шагов, в результа-

Таблица 6

Сравнительная эффективность различных программ оптимизации функции (32) при n=5; $\Delta x_0 = \Delta_0 = 1$

Номер программы	Шаг	Начальные значения x_{i0}	Делитель шага a	$\Delta x_0/\delta=10^5$			$\Delta x_0/\delta=10^3$		
				N	Время вычисления		N	Время вычисления	
					мин	с		мин	с
14/34	Абсолютный	0,1	2	234	57	11	155	38	34
			3	197	48	08	137	34	10
			4	156	38	07	90	22	23
			5	212	51	48	152	37	49
		0,555	2	235	57	25	139	34	49
			3	194	47	25	141	35	06
			4	186	45	27	115	28	06
			5	177	43	15	121	29	35
16/34	Относительный	1,0	3	196	47	40	147	36	52
			4	196	47	40	125	30	56
			5	194	47	20	129	32	20
17/34	Абсолютный	0,1		91	22	37	82	20	24
		0,555	10	171	42	32	132	32	48
18/34	Абсолютный	0,1	10	46	11	26	46	11	26
		0,555		171	42	32	132	32	48
19/34	Относительный	1,0	10	67	21	14	46	14	34
		0,555		90	28	32	66	21	00
		1,0	5	90	28	32	56	17	45
19/34	Абсолютный	0,1	10	79	25	03	56	17	45
		0,2		84	26	37	61	19	20
		0,555		130	41	13	107	33	55

те чего математическое ожидание среднего числа шагов может отличаться от среднего значения, полученного экспериментально. Анализ результатов испытания различных алгоритмов позволяет сделать вывод о наибольшей эффективности алгоритма, реализованного программой 19/34 с относительным шагом оптимизации.

Для функций, у которых экстремум расположен при значениях переменных, кратных начальным условиям, наиболее эффективен алгоритм, реализуемый программой 18/34. Но с увеличением эффективности программы растет занимаемый ею объем программной памяти, уменьшая количество адресов, по которым можно записать программу вычисления требуемой функции. Поэтому после анализа исследуемой функции, составления и оптимизации программы ее вычисления подсчитывают число адресов, которые могут быть заняты программой оптимизации, и выбирают последнюю из приведенных программ 14/34–19/34.

Однако рассмотренные стратегии могут не привести к успеху при исследовании многопараметрических функций, имеющих достаточно длинный спуск к глобальному экстремуму, не параллельный ни одной из координатных осей. В этом случае при поиске вдоль одной из координатных осей экстремум перемещается вдоль другой достаточно далеко и пропорциональное уменьшение шага не может быть признано оптимальным вследствие того, что нарушается допущение об определении экстремума с точностью до шага. Экстремум может сместиться на несколько шагов (иногда десятков шагов), и при кратном уменьшении шага придется делать слишком много итераций. Рассчитывать число шагов здесь безотносительно к исследуемой функции не имеет смысла.

Для таких функций может быть предложена другая стратегия последовательного спуска. После окончания цикла приближения по всем переменным с шагом Δx_i и перехода вновь к первой оптимизируемой переменной сравнивается значение функции до входа в цикл оптимизации и на выходе из цикла. Если значение функции изменилось, есть смысл повторить цикл без изменения шага оптимизации, в противном случае шаг оптимизации изменяют по формуле (26). Условием окончания вычислений может быть неравенство

$$|\Delta x_i| \leqslant \delta .$$

Программа 20/34. Оптимизация функции многих переменных с длинным пологим спуском к экстремуму, не параллельным ни одной из координатных осей, с кратным уменьшением шага:

ИПВ	КИП↑	×	ПА	ПП	29	ПА	ИПА	ПП	29
L0	00	№	П0	ИП8	ИПС	П8	–	x≠0	28
ИП8	ИП9	÷	ПВ	ИПD	–	x<0	00	С/П	КИП↑
+	КП↑	x≥0	47	x≠0	47	ПП	54	ИПС	·XY
ПС	–	x<0	07	вх	+	ПС	КИП↑	ИПА	/–/
+	КП↑	вх	В/О	...	...	...	...	В/О	

Содержание регистров памяти:

П8 – значение функции до входа в цикл приближения;

П9 – делитель шага a .

Остальные регистры памяти аналогичны предыдущим программам оптимизации.

В случае слишком длительной оптимизации по приведенной программе можно несколько преобразовать формулы изменения шага:

$$\Delta x_i = \Delta x_0 (1 - m/a) , \qquad (33)$$

где m – номер цикла вычисления $1 < m < a$; a – целое число $2 \leqslant a \leqslant 10$.

При этом циклы вычисления прекращают при условии $\Delta x_i = 0$.

Программа 21/34. Оптимизация функций многих переменных с пологим спуском к минимуму, не параллельным ни одной из координатных осей, с некратным уменьшением шага оптимизации:

ИПВ	КИП↑	×	ПА	ПП	27	ПА	ИПА	ПП	27
L0	00	№	П0	ИП8	ИПС	П8	–	x≠0	26
ИПВ	ИП9	–	ПВ	x=0	00	С/П	КИП↑	+	КП↑
x≥0	45	x≠0	45	ПП	52	ИПС	XY	ПС	–
x<0	07	вх	+	ПС	КИП↑	ИПА	/–/	+	КП↑
вх	В/О	...	...	...	...	В/О			

Если пользователь находит величину

$$\Delta x_i = \Delta x_{i-1} (1 - \frac{a-1}{a}) = \frac{\Delta x_{i-1}}{a} ,$$

полученную при останове программы 21/34, достаточной для точности вычисления оптимума функции, вычисление прекращается, если нет, то за Δx_i закрепляется значение $\Delta x_{i-1}/a$ и проводятся повторные циклы до выполнения условия

$$\Delta x_{i-1}/a^n \leqslant \delta .$$

Программа 22/34. Автоматическое изменение начального шага оптимизации и проверка его на допустимую погрешность приближения:

ИП9	ИП9	m	÷	П9	–	ПВ	ИПD	–	x<0
00									

Этот фрагмент программы вставляют в программу 21/34 между операторами с адресами 25 и 26. Естественно, что и ссылка на начало подпрограммы по адресам 05 и 09 изменяется с номера 27 на номер 38, а адрес условного перехода, размещенного по адресу 18, изменится с номера 26 на номер 37.

Последняя программа занимает в памяти микрокалькулятора 64 оператора, поэтому на вычисление функции остается всего 34 оператора.

Наиболее удобной для пологих и длинных непараллельных спусков считаем программу 21/34, а в случае возможности размещения – совокупность программ 21/34 и 22/34.

Все рассмотренные выше методы поиска экстремума функции относятся к сканирующим методам, поскольку с их помощью последовательно вычисляются значения функции вдоль одной из координатных осей, несмотря на то, что в некоторой исследуемой точке координатной плоскости может оказаться такое направление спуска, вдоль которого значение функции уменьшается в наибольшей степени.

Методы, с помощью которых определяют направление наискорейшего спуска и осуществляют перемещение по координатной плоскости в этом направлении, называют градиентными. Скорость минимизации с применением таких методов гораздо выше, чем с применением сканирующих методов. Однако градиентные методы требуют несравненно большего объема оперативной памяти, поэтому с помощью микрокалькулятора можно организовать только выбор направления шага и вычислить значение функции при этом шаге.

Рассмотрим алгоритм минимизации функции трех переменных методом наискорейшего спуска. Общее число всех положительных направлений спуска в этом случае — семь. Их удобно представить в двоичном коде, тогда код из чисел будет однозначно мнемонически определять направление шага (табл. 7).

Таблица 7

Коды направлений шагов

Код	Приращаемая переменная						
	x_1	x_2	$x_2; x_1$	x_3	$x_3; x_1$	$x_3; x_2$	$x_3; x_2; x_1$
Двоичный мнемонический	001	010	011	100	101	110	111
Десятичный	1	2	3	4	5	6	7

Последовательно, как в табл. 7, даем приращение соответствующим переменным, формируем из этих, а также из неизменяемых на конкретном шаге переменных стек и передаем его в подпрограмму вычисления функции. В этой же подпрограмме с помощью оператора КИП6 увеличиваем содержимое регистра П6 на единицу (формируем десятичный код направления).

На выходе из подпрограммы вычисления функции сравниваем новое значение с записанным в регистре памяти ПС. Если новое значение меньше предыдущего, записываем новое значение в регистр памяти ПС и закрепляем в регистре ПD код направления. Если новое значение функции больше предыдущего, оно пропадает, и содержимое регистра ПD, хранящего десятичный код направления, не изменяется.

Таким образом, после полного перебора приращений по всем направлениям в регистре ПD указан десятичный код направления перемещения вдоль координатных осей, а в регистре ПС — значение функции, вычисленной в этом направлении с шагом, хранимым в ПВ.

Если в регистре ПD после окончания цикла записан нуль, то принятое значение шага не уменьшило значения функции ни по одному из направления. Необходимо изменить шаг и повторить поиск градиента еще раз. В случае успешных шагов необходимо в регистры, соответствующие переменным, записать новые значения:

$$x_i = x_{i-1} + \Delta x_i,$$

причем i определяют мнемонически, переводя в двоичный код число, хранящееся в регистре ПD. Программа оптимизирована объединением фрагментов в подпрограммы, поэтому внешний вид ее не совпадает с описанием алгоритма.

Программа 23/34. Выбор направления приращения функции трех переменных по наибольшему градиенту:

3	ПП	44	ИП3	XY	ИП4	XY	ПП	61	4
ПП	44	ИП4	ИП2	ПП	61	ПП	53	ИП4	ПП
51	↻	ИП3	ПП	36	↻	ИП3	XY	ПП	51
ПП	36	XY	ПП	60	С/П	XY	ИП2	ПП	61
ПП	53	В/О	5	П1	КИП1	ИПВ	+	L0	45
В/О	ПП	60	3	П0	ПП	43	П9	↻	В/О
ИП9	...	...	КИП6	↻	ИПС	—	x<0	А	вх
+	ПС	ИП6	ПD	В/О					

Содержание регистров памяти:

П1 — счетчик номера переменной при косвенном обращении;

П2, П3, П4 — x_{01}, x_{02}, x_{03} соответственно;

П6 — регистр-счетчик кода направления;

П0, П9 — буферные регистры, П0=1;

ПВ — шаг оптимизации;

ПС — минимальное значение функции $F(x_i)$;

ПD — код направления наибольшего градиента.

Вместо многоточия в программу необходимо записать подпрограмму вычисления оптимизируемой функции, вместо А — адрес последнего оператора В/О, заканчивающего подпрограмму.

В приведенной программе переменные передаются в подпрограмму вычисления в стеке: x_1 — в операционном регистре X; x_2 — в регистре Z; x_3 — в операционном регистре Y. Такое расположение переменных в стеке вызвано удобством вычисления функции Розенброка. Если необходимо иное расположение переменных в стеке, можно по-иному сформировать стек.

Несколько сократить программу минимизации можно, использовав свободные регистры памяти в качестве буферных.

Программа 24/34. (сокращенная). Выбор направления приращения функции трех переменных по наибольшему градиенту:

ИП3	П2	ИП5	П4	ИП7	П6	ПП	55	ПС	Сх
П9	ПD	ПП	23	ИП5	П4	8	ПП	32	ПП
23	ИП9	С/П	ПП	31	6	ПП	32	ПП	31
В/О	4	П1	КИП1	ИПВ	+	КП1	ПП	51	ИПС
—	x<0	48	вх	+	ПС	ИПD	П9	ИП3	П2
В/О	ИПD	1	+	ПD	...	...	...	...	В/О

Содержание регистров памяти:

П1 — счетчик номера переменной при косвенном обращении;

П3, П5, П7 — начальные значения переменных x_{10}, x_{20}, x_{30} соответственно;

П2, П4, П6 — измененные значения переменных для обращения к ним в подпрограмме вычисления целевой функции x_{1i}, x_{2i}, x_{3i} соответственно;

П9 — код направления наибольшего градиента;

ПВ — шаг приращения;

ПD — формирователь десятичного кода перемещения.

Остальные регистры памяти не использованы.

В случае необходимости программу можно сократить за счет первых 12 операторов, но тогда перед началом работы необходимо в регистры П2, П4, П6 занести значения, соответственно размещенные в регистрах П3, П5, П7.

Для испытания алгоритмов минимизации функции используют функции Розенброка [17], так называемые "овражные" функции с отвесными склонами:

$$F(x_i) = 100(x_1^2 + x_2^2 - x_3)^2 + (x_1 - 1)^2 + (x_2 - 1)^2; \qquad (34)$$

$$F^*(x_i) = 0 \text{ при } x_1^* = x_2^* = 1;\ x_3^* = 2.$$

Программа 25/34. Сокращенная программа вычисления функции Резенброка с размещением переменных в стеке.

ПП	A_1	XY	–	XY	ПП	A_1	+	x^2	ИП5
×	+	+	В/О	1	XY	–	вх	x^2	XY
x^2	↻	В/О							

где A_1 – адрес начала подпрограммы (оператор 1).

Для испытания используем более короткую быстродействующую программу 24/34 с размещением переменных в регистрах памяти, а не в стеке.

Программа 26/34. Выбор направления поиска минимума функций Розенброка методом градиентного спуска:

ИП3	П2	ИП5	П4	ИП7	П6	ПП	55	ПС	Сх
П9	ПD	ПП	23	ИП5	П4	8	ПП	32	ПП
23	ИП9	С/П	ПП	31	6	ПП	32	ПП	31
В/О	4	П1	КИП1	ИПВ	+	КП1	ПП	51	ИПС
–	x<0	48	вх	+	ПС	ИПD	П9	ИП3	П2
В/О	ИПD	1	+	ПD	ИП2	x^2	ИП4	x^2	+
ИП6	–	x^2	ИП8	×	ИП2	1	–	x^2	+
ИП4	1	–	x^2	+	В/О				

Содержимое регистров памяти такое же, как в программе 24/34. Кроме того, в регистр П8 записано число 100.

При расчете по программе в регистре ПС сохраняется минимальное значение функции, которое получается при шаге в направлении наибольшего градиента.

При поиске минимума "овражных" функций градиентными методами желательно сразу выбрать достаточно большой шаг, так как при малом шаге на первых попытках алгоритм приводит "на дно оврага" и дальше следует "по дну оврага" весьма малыми шагами. Большой шаг позволяет быстрее найти зону глобального минимума, а затем спуститься "на дно оврага", к экстремуму.

Наилучшая стратегия заключается в проведении нескольких шагов от одинаковых начальных условий с кратным уменьшением шага. На каждом из шагов вычисления ведут до тех пор, пока имеется направление дальнейшего шага – регистр кода направления не равен нулю. Затем сравнивают значения функций, к которым пришли с различными шагами, и наименьшее принимают за локальный экстремум. От него проводят новые циклы итерации. Условием окончания вычислений является изменение функции на очередном шаге не выше заданной погрешности при значении шага, меньшем допустимого условиями задачи.

Для повышения достоверности результатов поиска на первых шагах нужно

в регистр, содержащий значение функции (ПС), занести это значение при начальных условиях приближения, причем в программе 24/34 это выполняется автоматически.

Последовательность решения задачи минимизации функции Резенброка показана в табл. 8. Очевидно, что градиентный метод поиска позволил получить глобальный экстремум за первые пять итераций, хотя было сделано еще дополнительно две итерации (13 и 14) для уточнения этого факта.

Продолжение вычислений после итерации 7, у которой второе минимальное значение функции $F(x_i) = 0{,}5$ через четыре итерации (15–18) приводит на "дно оврага" (строка 17) и далее следует по "дну оврага" с шагом 0,025.

Таблица 8

Последовательность итераций при минимизации функции Розенброка (34)

Номер итерации	Величина					Код направления	
	$F(x_i)$	x_3	x_2	x_1	Шаг Δx	Десятичный	Двоичный
	Регистры памяти						
	ПС	П7	П5	П3	ПВ	ПD	
1	2	0	0	0	3	0	000
2	2	0	0	0	2	0	000
3	2	0	0	0	1	5	101
4	1	1	0	1	1	6	110
5	0	2	1	1	1	0	000
6	2	0	0	0	0,5	7	111
7	0,5	0,5	0,5	0,5	0,5	0	000
8	2	0	0	0	0,25	1	001
9	1,95	0	0	0,25	0,25	7	111
10	1,2	0,25	0,25	0,5	0,25	6	110
11	0,5	0,5	0,5	0,5	0,25	6	110
12	0,5	0,75	0,75	0,5	0,25	0	000
13	0	2	1	1	10^{-2}	0	000
14	0	2	1	1	10^{-6}	0	000
15	0,5	0,5	0,5	0,5	0,1	5	101
16	0,42	0,6	0,5	0,6	0,1	6	110
17	0,36	0,7	0,6	0,6	0,1	0	000
18	0,36	0,7	0,6	0,6	0,025	4	100
19	0,32	0,725	0,6	0,6	0,025	5	101
20	0,30	0,750	0,6	0,625	0,625	6	110
21	0,29	0,775	0,625	0,625	0,025	5	101

На одну итерацию требуется 2–2,5 мин, построение всей таблицы заняло 45 мин.

Сканирующие методы для минимизации "овражных" функций непригодны вовсе, хотя следует заметить, что подобные функции в прикладных задачах технологии машиностроения встречаются крайне редко.

4. ПРИЛОЖЕНИЯ ПРОГРАММ ОПТИМИЗАЦИИ К ПРОЕКТИРОВАНИЮ РЕЖИМОВ МЕХАНИЧЕСКОЙ ОБРАБОТКИ

Выведем целевую функцию, используемую при расчете отдельных технологических операций механической обработки. Скорость резания определяют по эмпирическим формулам вида [5]

$$V = \frac{C_v k_v k_1 k_2 \ldots}{T^m s^{q_v} t^{x_v}}, \qquad (35)$$

где C_v – коэффициент скорости резания; k_v – коэффициент, зависящий от свойств обрабатываемого материала; k_1, k_2 – коэффициенты, зависящие от условий обработки; T – период стойкости инструмента, мин; m – показатель степени стойкости; s -- подача, мм/мин (или мм/об); q_v – показатель степени при подаче; t – глубина резания, мм; x_v – показатель степени при глубине.

При расчете различных технологических операций в формулу (35) могут входить и другие показатели.

Определив скорость резания, определяем частоту вращения шпинделя, об/мин:

$$n = \frac{1000V}{\pi d}. \qquad (36)$$

где d – диаметр обработки (инструмента), мм.

Машинное время определяют по формуле

$$t_м = i\frac{l + l_1}{s_o n}k, \qquad (37)$$

где $i = z/t$; i – число проходов; z – припуск, мм; t – глубина резания, мм; l – длина поверхности обработки, мм; l_1 – длина на врезание и перебег, мм; s_o – подача на один оборот шпинделя, мм/об; k – коэффициент, учитывающий ускоренные обратные ходы (k = 1,05 ... 1,1).

Основное время на обработку детали определится как сумма машинного и вспомогательного времени. Вспомогательное время $t_{всп}$ включает, кроме прочих, время на снятие, переточку, установку и настройку инструмента $t_{зат}$. Поскольку считаем, что при оптимизации изменению подвержено только время $t_{зат}$, постоянную часть $t_{всп}$ в расчет принимать не будем. Определим переменную часть $t_{всп}$, приходящуюся на одну деталь.

Число деталей N и число переточек инструмента $N_{пер}$ за смену на оптимизируемой операции

$$N = t_{см}/t_o, \; N_{пер} = t_{см}/T, \qquad (38)$$

где $t_{см}$ = 480 мин – продолжительность смены; t_o – основное время обработки детали;

$$t_o = t_м + t_{всп};$$

$$t_{всп} = t_{зат}N_{пер}/N = t_o t_{зат}/T, \qquad (39)$$

где $t_м$ – машинное время обработки детали.

В случае централизованной заточки инструмента вместо времени заточки

необходимо брать время замены инструмента, так как время заточки совмещено с цикловым временем обработки деталей.

Исходя из формул (38) и (39), можно получить

$$t_o = t_м + t_o t_{зат}/T,$$

откуда

$$t_o = t_м/(1 - t_{зат}/T). \qquad (40)$$

Следовательно, целевая функция запишется как последовательность формул (35)–(37), (40). Ее можно записать и по более сложным формулам [5]. Рассмотрим конкретные программы оптимизации для различных режимов обработки.

Оптимизация режимов резания при точении. Формула скорости резания (35) при точении имеет вид [5]

$$V = \frac{C_v k_v}{T^m t^{x_v} s^{y_v}}.$$

Формулы (36), (37) и (40) остаются неизменными.

В качестве программы оптимизации принимаем программу 19/34 с абсолютным шагом оптимизации, поскольку целевая функция имеет экстремум по одной переменной T.

Программа 27/34. Оптимизация режима резания при точении:

ПП	52	ПС	1	ИПВ	X	ПА	ПП	27	ПА
ИПА	ПП	27	L0	03	1	П0	ИПВ	ВП	1
/–/	ПВ	ИПD	–	x<0	03	С/П	КИП↑	+	КП↑
x≥0	45	x≠0	45	ПП	52	ИПС	XY	ПС	–
x<0	10	вх	+	ПС	КИП↑	ИПА	/–/	+	КП↑
вх	В/О	ИП2	ИП1	x^y	ИП5	ИП4	x^y	XY	↻
X	ИП3	XY	÷	ИП6	÷	π	÷	ИП7	XY
÷	ИП4	÷	1	ИП8	ИП1	÷	–	÷	В/О

Содержание регистров памяти:

П1 – период стойкости инструмента T; мин;

П2 – показатель степени m;

П3 – $C_v K_v/t^{x_v}$;

П4 – подача на оборот шпинделя s_o, мм/об;

П5 – показатель степени y_v;

П6 – $d/1000$ – диаметр детали d, мм;

П7 – $i(l + l_1)k$;

П8 – время заточки и установки инструмента $t_{зат}$, мин;

П0 – регистр-счетчик переменных в цикле;

ПА – буферный регистр для хранения шага в процессе расчета;

ПВ – шаг оптимизации;

ПС – минимальное значение целевой функции, мин;

ПD – точность приближения.

Поскольку в целевой функции всего один параметр оптимизации и работу ведем с абсолютным шагом оптимизации, операторы с адресами 03, 05, 13–16

могут быть исключены. Операторы косвенного обращения КИП↑ и КП↑ могут быть заменены операторами прямого обращения ИП1, П1, благодаря этому можно сэкономить шесть адресов программ и один регистр памяти.

Программа 28/34. Упрощенная программа оптимизации режима резания при точении по параметру "период стойкости инструмента Т":

ПП	46	ПС	ИПВ	ПА	ПП	21	ПА	ИПА	ПП
21	ИПВ	ВП	1	/–/	ПВ	ИПD	–	x<0	03
С/П	ИП1	+	П1	x≥0	39	x≠0	39	ПП	46
ИПС	XY	ПС	–	x<0	08	вх	+	ПС	ИП1
ИПА	/–/	+	П1	вх	В/О	ИП2	ИП1	x^y	ИП5
ИП4	x^y	XY	↻	×	ИП3	XY	÷	ИП6	÷
π	÷	ИП7	XY	÷	ИП4	÷	1	ИП8	ИП1
÷	–	÷	В/О						

Содержание регистров памяти такое же, как и в программе 27/34, за исключением свободного регистра П0.

Среднее время оптимизации по этим двум программам 8–15 мин. После расчета в регистре П1 записан оптимальный период стойкости инструмента T, в регистре ПС – наименьшее основное время. Для вычисления частоты вращения шпинделя в оборотах в минуту необходимо в программе 28/34 по адресу 62 записать оператор С/П и нажать клавиши В/О и С/П. Спустя 1 мин на табло высвечивается частота вращения шпинделя. Скорость резания при оптимальных параметрах определяется аналогично по адресу 58.

В качестве примера рассмотрим задачу: $T = 20$ мин; $m = 0{,}2$; $C_v k_v / t^{x_v} = 36 \cdot 1{,}1/3^{0{,}45} = 24{,}15$; $s_0 = 0{,}2$ мм/об; $y_v = 0{,}2$; $d/1000 = 0{,}08$; $i(l+l_1)k = 1(100+12)1 = 112$ мм; $t_{зат} = 15$ мин; $\Delta T_0 = \Delta X_0 = 10$ мин; $\delta = 10^{-1}$ мин; $a = 10$.

Время вычисления 5,2 мин; оптимальный период стойкости $T_{опт} = 90{,}0$ мин; минимальное основное время $t_0^* = 12{,}47$ мин; скорость резания $V = 13{,}5$ м/мин, частота вращения шпинделя $n = 53{,}9$ об/мин. Для $t_{зат} = 12$ мин время вычисления составило 8 мин; $T_{опт} = 72$ мин; $t_0^* = 11{,}9$ мин; $V = 14{,}17$ м/мин; $n = 56{,}4$ об/мин.

Оптимизация режимов резания при сверлении. Формула скорости резания (35) при сверлении в зависимости от параметров процесса записывается [5]

$$V = \frac{C_v d^{x_v}}{T^m s_0^{y_v}}.$$

Частоту вращения шпинделя можно вычислить по формуле (36), а производительность обработки по формуле

$$t_м = \frac{l + l_1}{s_0 n}.$$

В случае глубокого сверления, когда длина обработки превышает $(5 \ldots 8)d$, необходимо несколько раз вынимать сверло, поэтому формула имеет вид

$$t_м = \frac{l + il_1}{s_0 n} k,$$

где k – коэффициент, зависящий от числа ускоренных вводов-выводов и глубины сверления.

Формула основного времени не отличается от формулы (40).

В качестве оптимизируемых параметров примем период стойкости инструмента T и осевую подачу s_0. Допустим, что максимальная подача, ограниченная мощностью станка для принятого типа станка, не должна превышать 0,2 мм/об. В качестве программы оптимизации принимаем программу 19/34 с ограничениями $s_0 \leqslant 0{,}2$ мм/об. Шаг оптимизации принимаем относительный, так как величины T и s_0 отличаются на четыре порядка. Оптимизацию можно начинать с величин $T_0 = 20 \div 50$ мин; $s_0 = 0{,}02 \div 0{,}05$ мм/об. Начальный относительный шаг оптимизации $\Delta_0 = 1$.

Программа 29/34. Оптимизация режима сверления по параметрам: период стойкости инструмента T и осевая подача инструмента s_0:

ПП	57	ПС	КИП↑	ИПВ	×	ПА	ПП	27	ПА
ИПА	ПП	27	L0	03	2	П0	ИПВ	ВП	1
/–/	ПВ	ИПD	–	x<0	03	С/П	КИП↑	+	КП↑
x≥0	51	x≠0	51	5	1/x	ИП2	–	x≥0	51
ПП	58	ИПС	XY	ПС	–	x<0	10	вх	+
ПС	КИП↑	ИПА	/–/	+	КП↑	вх	В/О	ИП4	ИП1
x^y	ИП3	XY	÷	ИП5	ИП2	x^y	XY	↻	÷
ИП6	÷	ИП7	XY	ИП2	×	÷	1	ИП8	ИП1
÷	–	÷	В/О						

Содержание регистров памяти:

П1 – период стойкости сверла T, мин;

П2 – осевая подача шпинделя s_0, мм/об;

П3 – $C_v d^{x_v}$, мм;

П4 – показатель степени m;

П5 – показатель степени y_v;

П6 – $\pi d/1000$, d, мм;

П7 – длина обработки $l + l_1$ или $(l + l_1)k$, мм;

П8 – время заточки и установки сверла $t_{зат}$, мин.

Содержание остальных регистров памяти такое же, как и в программе 27/34.

Проверка подачи s_0 на ограничение сверху введена в программу по адресам 35–41. Величину 0,2 получаем в программе с помощью операторов 5 и 1/x тем, чтобы не использовать регистр памяти.

В качестве примера рассмотрим задачу: $d = 8$ мм; $s_0 = 0{,}05$ мм/об; $T = 60$ мин; $l + l_1 = 40$ мм; $t_{зат} = 20$ мин; $C_v = 5$; $x_v = 0{,}4$; $y_v = 0{,}7$; $m = 0{,}2$; $a_0 = 1$; $\delta = 10^{-2}$; $s_{max} = 0{,}2$ мм/об. Число оптимизируемых параметров – 2, поэтому в программе 29/34 запишем вместо № число 2. Это же число запишем в регистр П0.

Время счета составило 5 мин; $T_{опт} = 120$ мин; $t_0^* = 0{,}44$ мин; $n_{опт} = 543$ об/мин; $s_{опт} = 0{,}2$ мм/об. Следовательно, характер зависимости целевой функции – производительности сверления от подачи не имеет экстремумов. Оптимальная подача ограничена сверху максимальным значением

$$s_{опт} = s_{max} = 0{,}2 \text{ мм/об}.$$

Оптимизация режимов резания при фрезеровании. Формула скорости резания (35) при фрезеровании имеет вид [5]

$$V = \frac{C_V D^{g_V} k_V}{T^m t^{x_V} s^{y_V} B^{u_V} z^{p_V}}.$$

Частота вращения шпинделя описывается формулой (36), а машинное время обработки

$$t_м = i\frac{l + l_1}{s_z z n}.$$

Фрезы, в отличие от других режущих инструментов, имеют большое число режущих зубьев, поэтому время снятия заточки и установки фрезы будет зависеть от числа зубьев фрезы

$$t_{зат} = t_{0\,зат} + t_1 z, \qquad (41)$$

где $t_{0\,зат}$ – время на установку фрезы и подналадку станка; t_1 – время на заточку одного зуба.

При анализе целевой функции очевидно, что при стремлении z к нулю будет неограниченно возрастать машинное время, а при стремлении z к некоторому большому значению увеличится вспомогательное время, поэтому представляется возможным отыскать экстремум производительности, оптимизируя функцию по z.

Дальнейший анализ целевой функции позволяет заметить, что изменение одного параметра оптимизации z существенно влияет на смещение минимума функции по второй переменной T, поэтому стратегия последовательного спуска с кратным уменьшением шага при переходе к новому циклу расчетов может не привести к успеху. По крайней мере время расчета будет составлять несколько часов. Здесь уместно применить программу 21/34, совместив ее с программой 22/34. Но в этом случае слишком много регистров памяти занимает программа приближения. Высвободить один регистр можно программно задавая квадрат наименьшего шага приближения. Это осуществляется операторами К, 10^x, 1/x или К, /–/, 10^x, где К – порядок наименьшего шага.

Программа 30/34. Оптимизация режима фрезерования по параметрам z – число зубьев фрезы и T – период стойкости инструмента:

ПП	67	ПС	ИПВ	КИП↑	×	ПА	ПП	37	ПА
ИПА	ПП	37	L0	03	2	П0	ИПВ	ИП9	–
ПВ	x=0	03	ИП9	ИП9	5	÷	П9	–	ПВ
2	/–/	10^x	–	x<0	03	С/П	КИП↑	+	КП↑
x⩾0	60	x≠0	60	ИП2	ИПD	–	x<0	60	ПП
67	ИПС	XY	ПС	–	x<0	10	вх	+	ПС
КИП↑	ИПА	/–/	+	КП↑	вх	В/О	ИП4	ИП1	x^y
ИП5	ИП2	x^y	XY	↻	×	ИП3	×	ИП2	÷
1	ИП2	t_1	×	ИП8	+	ИП1	÷	–	÷
В/О									

Содержание регистров памяти:

П0 – число оптимизируемых переменных (в данной программе – 2);

П1 – период стойкости инструмента T, мин;

П2 – число зубьев фрезы z;

П3 – коэффициент, учитывающий все неизменяемые величины формул (35), (36), (37) с тем, чтобы сократить время вычислений и разгрузить регистры памяти;

П4 – показатель степени при стойкости m;

П5 – показатель степени при числе зубьев p_V;

П8 – время на установку фрезы и подналадку станка $t_{0\,зат}$;

П9 – уменьшаемое шага (0,1–0,5);

ПВ – относительный шаг оптимизации;

ПD – максимально допустимое число зубьев фрезы z_{max}.

На адресе 25 записано число 5, которое уменьшает в 5 раз шаг оптимизации. По адресу 82 вместо t_1 необходимо поставить требуемое значение времени заточки одного зуба фрезы.

В качестве примера рассмотрим задачу: D = 45 мм; B = 50 мм; t = 3 мм; T_0 = 100 мин; $l + l_1$ = 160 мм; $t_{0\,заг}$ = 12 мин; z_0 = 8; x_V = 0,38; y_V = 0,28; u_V = 0,08; p_V = 0,1; g_V = 0,17; C_V = 70,0; s_z = 0,03 мм/зуб; m = 0,33; t_1 = = 2 мин; z_{max} = 50. Число оптимизируемых переменных – 2. Время расчета 18–25 мин. Результаты расчета: $T_{опт}$ = 450,6 мин; $z_{опт}$ = 50; t_0^* = 1,297 мин.

Изменение начальных условий в части времени заточки одного зуба с двух до трех минут и показателя степени при числе зубьев p_V до 0,3 привело к другому результату: $T_{опт}$ = 651,4 мин; $z_{опт}$ = 50; t_0^* = 3,207 мин.

Необходимо отметить, что задача оптимизации режима фрезерования несколько упрощена. Для получения более точных результатов следует учесть зависимость числа зубьев фрезы от ее диаметра, найти более сложную целевую функцию производительности от числа зубьев и даже ввести третью оптимизируемую переменную – диаметр фрезы. Однако нужно так распределить регистры памяти, чтобы программы оптимизации и нахождения целевой функции уместились в программную память микрокалькулятора.

Приведенные примеры расчетов режимов резания показывают, что можно использовать оптимизационные программы при работе на микрокалькуляторах, либо разработать иные виды целевых функций. Важно помнить, что выбору программы оптимизации должны предшествовать построение и анализ целевой функции, творческая и аналитическая стадии. Микрокалькулятор выполнит всю рутинную расчетную работу тем быстрее, чем лучше будет выбрана прикладная программа оптимизации.

ГЛАВА IV.

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

Решение любой прикладной задачи математическими методами, вообще говоря, есть математическое моделирование, поскольку, записав условие задачи с помощью математических выражений, мы по существу строим математическую модель реального физического процесса, а разрешив математическое выражение относительно интересующих нас неизвестных, получаем те значения, которые исследуемая величина принимает в реальных условиях. Под моделью

понимают систему, не отличимую от моделируемого объекта в отношении некоторых свойств, полагаемых "существенными", и отличимую от него по всем остальным свойствам, которые полагают "несущественными". При этом отсутствие в модели "несущественных" элементов не менее важно, чем присутствие в ней "существенных" (Нюрберг Н.Д. О познавательных возможностях моделирования// Математическое моделирование жизненных процессов. – М., 1968). Естественно, что перед тем как сделать окончательный вывод о переносе результатов математического моделирования в реальные условия, необходимо решить вопрос об адекватности модели, т. е. о подобии "существенных" свойств математической модели таким же свойствам реального физического объекта в некоторой серии натурных экспериментов.

При составлении и решении математических выражений возникает необходимость делать какие-то допущения относительно некоторых параметров физического объекта. Это те параметры, которые изменяются либо незначительно и, по нашему мнению, на конечный результат не влияют, либо перегружают модель мелкими уточнениями, не позволяющими делать обобщения и распространить свойства модели на широкий круг аналогичных явлений, либо потому, что эти параметры пока невозможно учесть и описать. Поэтому всякая модель имеет определенную степень приближения и ошибки, которые могут быть уменьшены при сравнении поведения модели и реального физического объекта при проведении натурных испытаний. Более того, в процесс построения модели включают этап экспериментальной проверки, на котором модель уточняется и совершенствуется.

Математическое моделирование в последнее время необычайно широко распространено и внедряется во все области техники. Это объясняется тем, что процесс автоматизированного проектирования немыслим без построения моделей (в настоящее время широко разрабатываются вопросы автоматического построения математических моделей), а это, в свою очередь, требует формального описания реальных процессов и позволяет применить для моделирования отлично разработанный математический аппарат.

Наиболее широко математические модели применяют когда натурный эксперимент громоздок, дорог или даже невозможен. Математическую модель строят также тогда, когда необходимо проанализировать большое число факторов. Если в процессе исследования математическая модель подтверждает наблюдаемые физические явления, ее можно экстраполировать на более широкую область. Однако создание адекватных математических моделей – это искусство, поскольку из тысяч наблюдаемых в опыте факторов необходимо отобрать самые весомые, выяснить и описать их влияние на поведение модели. Это может сделать только хороший математик, тонко чувствующий физику процесса.

Все реальные механические колебательные процессы, к которым можно отнести и обработку деталей на металлорежущих станках, описываются дифференциальными уравнениями и их системами. Однако сложный характер зависимостей при металлообработке не позволяет применить для моделирования процессов линейные дифференциальные уравнения, для которых разработан большой аналитический математический аппарат. Сложности, возникающие при анализе нелинейных систем, приводят к тому, что исходное нелинейное

уравнение линеаризуют различными методами (например, кусочно-линейной аппроксимацией, методом гармонического баланса и другими [1, 2]); при этом нередко возникают допущения, которые существенно идеализируют математическую модель, отрывают ее от реального процесса. Другими словами, при линеаризации исходной нелинейной динамической модели могут фактически остаться незамеченными эффекты, присущие только нелинейным системам. Поэтому в настоящее время все большее распространение получает моделирование с помощью ЭВМ, при котором используют различные численные методы нахождения решений и исследования дифференциальных уравнений. Некоторые задачи динамики и колебаний с успехом могут быть решены на микрокалькуляторах. Составление и анализ дифференциальных уравнений рассматриваются в теории колебаний [1] и теории автоматического управления и регулирования [2]. Здесь рассмотрим только некоторые численные методы решения дифференциальных уравнений и построение их фазовых траекторий с использованием программируемых микрокалькуляторов.

1. ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Численные методы решения дифференциальных уравнений с различной степенью точности достаточно подробно разработаны и описаны [8, 11–13, 37], поэтому здесь рассмотрим только два из них.

Метод Рунге–Кутта второго порядка точности (усовершенствованный метод Эйлера). Для дифференциального уравнения

$$\dot{x} = f(x, t), \tag{42}$$

используя начальные условия x_0; t_0, можно вычислить последующие значения функции $x = x(t)$ по формулам [12]

$$k = x_i + \frac{h}{2} f(x_i; t_i); \tag{43}$$

$$x_{i+1} = x_i + hf(k; t_i + h/2), \tag{44}$$

где $i = 0, 1, 2, ...$; h – шаг интегрирования.

Глобальная погрешность имеет второй порядок относительно шага интегрирования.

Программа 31/34. Численное интегрирование дифференциального уравнения первого порядка методом Рунге–Кутта второго порядка точности (усовершенствованным методом Эйлера):

ПП	24	ИП1	П0	+	П1	ИП2	ИП3	+	П2
ПП	24	2	×	ИП0	+	П1	ИП2	ИП3	+
П2	С/П	БП	00	...	...	...	ИП3	×	В/О

Содержание регистров памяти:

П0 – буферный регистр для хранения x_i;

П1 – x_0 в момент пуска, затем x_i;

П2 – t_0 в момент пуска, затем t_i;

П3 – половина шага интегрирования $h/2$.

Вместо многоточия в программу записывают подпрограмму вычисления функции (42) с использованием регистров П1 и П2. Остальные регистры памяти можно использовать для хранения коэффициентов уравнения (42).

В качестве примера рассмотрим функцию [39]

$$\dot{x} = t\sqrt{x}$$

с начальными условиями $x_0 = 1$ и $t_0 = 1$ и шагом интегрирования $h = 0{,}1$. Микрокалькулятор выдает последовательно: 1,1076; 1,2317; 1,3742 и т. д., точные значения: 1,1078; 1,2321; 1,3748.

Наибольший интерес в механических колебательных системах представляют процессы, связанные с учетом инерционных сил, которые описываются дифференциальными уравнениями второго порядка:

$$\ddot{x} = f(\dot{x}; x; t), \tag{45}$$

которое можно представить как систему из двух уравнений первого порядка:

$$\dot{x} = y;$$

$$\dot{y} = f(y; x; t). \tag{46}$$

Для численного решения этой системы можно применить усовершенствованный метод Эйлера (43), (44). Программа решения таких систем имеется [37]; она приведена в приложении для уравнения (45), записанного в виде системы (46).

Программа 32/34. Решение дифференциального уравнения второго порядка модифицированным методом Эйлера:

ПП	24	ИП3	+	П3	ИПА	2	×	ПА	ПП
24	2	÷	ПА	ИП3	+	П3	ИП2	П5	ИП1
П4	С/П	БП	00	...	...	...	ИПА	×	ИП4
+	П1	ИП2	ИПА	×	ИП5	+	П2	ИПА	В/О

Содержание регистров памяти:

П1, П4 – x_0 – перед пуском программы начальное условие, затем очередное значение переменной x_i. При обращении в подпрограмме нужно использовать только П1.

П2, П5 – перед пуском программы начальное условие $\dot{x}_0$, затем очередное значение скорости изменения переменной; при обращении в подпрограмме нужно использовать только П2;

П3 – начальное условие независимой переменной t_0, затем t_i;

ПА – половина шага интегрирования $h/2$.

Остальные регистры памяти не загружены и могут быть использованы для хранения коэффициентов уравнения.

Вместо многоточия в программу записывают подпрограмму вычисления функции в виде (45) или, что то же самое, в виде второго уравнения системы (46). Первое уравнение системы (46) в данной программе представлено простым вызовом скорости $\dot{x}$ (оператор ИП2 по адресу 32). Если в первом уравнении системы (46) также имеется функциональная зависимость, можно в программе 32/34 вместо оператора с адресом 32 вставить подпрограмму вычисления требуемой функции.

Для примера рассмотрим уравнение [39]

$$\ddot{x} = \frac{3}{4}\left(\frac{\dot{x}t}{\sqrt{x}} + 1\right)$$

с начальными условиями $t_0 = x_0 = \dot{x}_0 = 1$. Шаг интегрирования $h = 0{,}05$.

Подпрограмма вычисления этой функции имеет вид

ИП3	ИП2	×	ИП1	$\sqrt{x}$	÷	1	+	3	×
4	÷								

Время одного цикла вычислений 20–25 с.

Выполняя четыре итерации, получим

t_1	...	1,0	1,05	1,10	1,15	1,20
x_i	...	1,0	1,076	1,153	1,236	1,320
$\dot{x}_i$	...	1,0	1,051	1,105	1,162	1,221
x_i точн.	...	1,0	1,052	1,108	1,168	1,232

Приведенные методы решения дифференциальных уравнений просты, но имеют невысокую точность вычислений, убывающую с каждым циклом итераций. Точность можно повысить за счет уменьшения шага интегрирования, но при этом существенно возрастает время вычислений, и при слишком малом шаге погрешность может даже повыситься из-за накопления ошибок округления в микрокалькуляторе. Поэтому, если позволяет программная память микрокалькулятора, используют численные методы интегрирования дифференциальных уравнений более высоких порядков. Наибольшее распространение нашел метод Рунге–Кутта четвертого порядка.

Метод Рунге–Кутта четвертого порядка точности. Суть метода заключается в следующем. При известных начальных условиях x_0 и t_0 уравнения (42) несколько раз на шаге вычисляют скорость изменения функции. Затем ее усредняют и переходят к новой точке, которую принимают за начальную. Этот метод имеет глобальную погрешность не ниже четвертого порядка относительно шага интегрирования. Оценку погрешности приведем ниже.

Формулы для нахождения значений x_i [9]:

$$x_{i+1} = x_i + \Delta x_i; \tag{47}$$

$$\Delta x_i = 1/3\,[k_1 + 2(k_2 + k_3) + k_4], \tag{48}$$

где $i = 0;\ 1;\ 2, \ldots;$

$$k_m = (h/2) f(x^m; t^m),$$

$$m = 1;\ 2;\ 3;\ 4;$$

$$x^1 = x_i;\ t^1 = t_i;$$

$$x^2 = x^1 + k_1;\ t^2 = t^1 + h/2;$$

$$x^3 = x^1 + k_2;\ t^3 = t^1 + h/2;$$

$$x^4 = x^1 + 2k_3;\ t^4 = t^1 + h.$$

Приведенные формулы легко запрограммировать, причем здесь выгодно использовать преимущество языка программирования микрокалькуляторов,

заключающееся в том, что к подпрограммам можно обращаться начиная с различных операторов. Это существенно экономит место в программной памяти микрокалькулятора.

Программа 33/34. Решение дифференциального уравнения (42) методом Рунге–Кутта четвертого порядка:

ПП	39	ПП	29	2	×	+	ПП	34	2
×	ПА	+	ПП	29	+	3	÷	ИПВ	+
ПВ	П1	L0	00	С/П	№	П0	БП	00	ИП2
ИП3	+	П2	↻	ИПА	ИПВ	+	П1	↺	...
...	ИП3	×	ПА	В/О					

Содержание регистров памяти:

П1, ПВ – первоначально начальные значения x_0, в процессе расчета x_i;

П2 – независимая переменная t_0, в процессе расчета t_i;

П3 – половина шага интегрирования $h/2$;

П0 – счетчик циклов вычислений;

ПА, ПВ – буферные регистры.

Остальные регистры памяти не используются. В приведенной программе реализован программный останов через определенное число циклов с тем, чтобы не выводить на индикацию промежуточные вычисления и работать с небольшим шагом. Вместо многоточия в программу заносят подпрограмму вычисления заданной функции (42). Вместо знака № в программу записывают требуемое число циклов, через которое необходимо выдать результат. В некоторых случаях выгодно этот оператор опустить, а после программного останова в регистр X вводить требуемое число циклов вычислений до индикации. Тогда в зависимости от характера функции оператор произвольно будет изменять шаг построения таблицы функции или графика.

Если исследуемая функция достаточно сложная и не умещается в программной памяти микрокалькулятора, операторы с адресами 22–26 можно исключить. Приращение Δx_i в этой программе формируется в стековом регистре, поэтому в подпрограмме выключены операторы поворота стека с адресами 33; 38.

В качестве примера расчета использована тестовая функция к программе 31/34. Подпрограмма вычисления функции имеет вид

ИП1 $\sqrt{x}$ ИП2 ×

Результаты расчета с точностью до четвертого знака совпадают с теоретическим значением решения. Время одного цикла вычислений составляет 23 с.

Учитывая то, что дифференциальные уравнения второго порядка в механике имеют практическую важность, для их численного интегрирования с использованием метода Рунге–Кутта четвертого порядка разработаны специальные формулы [13]:

$$x_{i+1} = x_i + 2\dot{x}_i h/2 + \Delta x_i; \tag{49}$$

$$\dot{x}_{i+1} = \dot{x}_i + \Delta\dot{x}_i, \tag{50}$$

где $i = 0, 1, 2...$

$$\Delta x_i = \frac{2}{3}\,\frac{h}{2}\,(k_1 + k_2 + k_3); \tag{51}$$

$$\Delta\dot{x}_i = \frac{1}{3}\,[k_1 + 2(k_2 + k_3) + k_4]; \tag{52}$$

$$k_m = \frac{h}{2} f(\dot{x}^m; x^m; t^m);$$

$$m = 1;\ 2;\ 3;\ 4;$$

$$\dot{x}^1 = \dot{x}_i;\ x^1 = x_i;\ t^1 = t_i;$$

$$\dot{x}^2 = \dot{x}^1 + k_1;\ x^2 = x^1 + \dot{x}^1 h/2;\ t^2 = t^1 + \frac{h}{2};$$

$$\dot{x}^3 = \dot{x}^1 + k_2;\ x^3 = x^1 + \dot{x}^1\frac{h}{2} + k_1\frac{h}{2};\ t^3 = t^1 + h/2;$$

$$\dot{x}^4 = \dot{x}^1 + 2k_3;\ x^4 = x^1 + \dot{x}^1 h + k_2 h;\ t^4 = t^1 + h.$$

Априорная оценка точности вычислений в зависимости от шага [36]

$$\frac{k_3 - k_2}{k_1 - k_2} \leqslant 0{,}01n, \text{ где } 1 < n < 10$$

однозначного ответа о точности решения на данном шаге не дает. Других надежных априорных оценок нет, поэтому пользуются апостериорными оценками, например, формулой Рунге [22]:

$$\Delta\dot{x}(x, t, h) \approx \frac{\dot{x}(x, t, h) - \dot{x}(x, t, h/2)}{2^K - 1}, \tag{53}$$

где K – порядок точности способа.

При использовании формулы Рунге (53) оказывается удобным оценивать точность вычисления при кратном изменении шагов.

Наиболее короткую программу решения дифференциальных уравнений второго порядка методом Рунге–Кутта четвертого порядка, предложенную в [39], приводим здесь без изменения. Отметим, что она построена на решении системы из двух уравнений первого порядка аналогично (46).

Программа 35/34. Решение дифференциального уравнения второго порядка методом Рунге–Кутта четвертого порядка [39]:

В/О	3	÷	П9	ПВ	ИПС	3	÷	П8	ПА
С/П	ПП	27	ПП	23	ПП	45	ПП	27	ИПВ
ИПА	ПП	38	ИП6	ИП7	+	П6	ПП	54	ИП7
×	П4	ИПВ	ИП7	×	П5	ИП9	ИП8	ИП5	+
ПА	XY	ИП4	+	ПВ	ИПС	ИП5	+	ПС	ИПD
ИП4	+	ПD	В/О	...	...	...	В/О		

Содержание регистров памяти:

П4, П5, П8, П9, ПС, ПD – буферные регистры;

П6 – независимая переменная t_i;

П7 – половина шага интегрирования $h/2$;

ПА – значение искомой функции x_i;

ПВ – значение производной искомой функции $\dot{x}_i$.

Первоначально, перед пуском программы в регистры ПС и ПD заносят утроенные значения начальных условий x_0 и $\dot{x}_0$ соответственно.

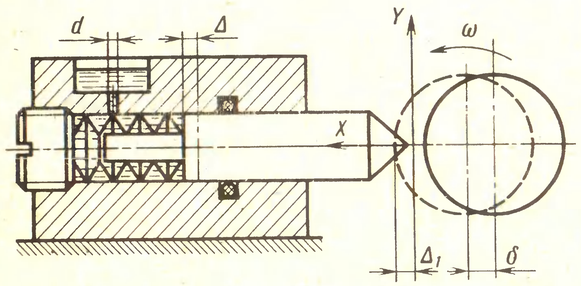
В регистр П6 вносят значения начальных условий независимой переменной, в регистр П7 – половину шага интегрирования. Содержимое регистра ПD вводят в операционный регистр X. Результат вычисления получают в регистрах стека: x_i – в операционном регистре X, $\dot{x}_i$ – в операционном регистре Y.

Интересная особенность этой программы состоит в том, что начинается она оператором В/О, который при незаполненном адресном стеке передает управление по адресу 01. Его применение вызвано другим оператором В/О, стоящим по адресу 53. Второй оператор может либо заканчивать подпрограмму и возвращать управление на адрес, с которого было обращено к подпрограмме, либо на начало программы, если завершается выполнение основной программы.

Здесь видно, как изобретательно составлена весьма сложная программа с использованием возможностей адресного стека. Однако при работе с ней необходимо помнить, что если прервать вычисления (например, в процессе редактирования и отладки программы), адресный стек запоминает адрес, с которого передано управление на подпрограмму, выполнение которой прервано, поэтому после нажатия клавиш В/О и С/П программа не обязательно начнет работать с оператора 01, а предварительно считает число, хранимое в адресном стеке. Избежать неконтролируемого выполнения операций в процессе отладки можно только дождавшись программного останова машины. Если все же адресный стек заполнится каким-либо числом – адресом возврата с неотработанной до конца подпрограммы, – микрокалькулятор при повторных пусках начнет зацикливаться, выдавать в месте перехода на подпрограммы индикацию ошибки или неверно считать. От этого можно избавиться, последовательно нажав клавиши В/О и ПП 3–4 раза. Приведенная программа занимает много адресных регистров памяти в качестве буферных, оставляя для коэффициентов уравнения только регистры П0, П2, П3, П4.

Приложение численного решения дифференциальных уравнений при моделировании процесса механической обработки. В качестве примера рассмотрим дифференциальное уравнение, описывающее процесс поверхностного пластического деформирования (ППД) детали, укрепленной в центрах с биением $\delta = 0{,}05$ мм (рис. 19). Для приведенной расчетной схемы может быть составлено дифференциальное уравнение

$$m\ddot{x} = -C\dot{x} - K(\Delta + x + \delta \sin \omega t) + Es(\Delta_1 - x - \delta \sin \omega t)\,\mathrm{sgn}(\Delta_1 - x - \delta \sin \omega t), \qquad (54)$$



где m – приведенная масса инструмента, кг; C – коэффициент демпфирования, Н/(мм/с);

Рис. 19. Схема динамического моделирования процесса ППД с гидравлическим демпфером

K – жесткость пружины, Н/мм; Δ – предварительный натяг пружины, мм; δ – биение детали, мм; ω – угловая скорость детали, рад/с; E – жесткость материала детали на единицу площади деформирования, Н/мм^3; s – площадь пятна контакта деталь–инструмент, мм^2; Δ_1 – предварительный натяг инструмента в деталь, мм; sgn( ) – единичная функция, описывающая силовое воздействие поверхности детали на инструмент в случае контакта и отсутствие его при отрыве инструмента от детали.

Уравнение (54) может быть решено и аналитически, но здесь приведены некоторые изменения модели, которые при аналитических решениях учесть сложно.

Программа 36/34. Подпрограмма вычисления значения функции (54): До адреса 54 вводят программу 35/34.

ИП6	2	10^x	×	sin x	ИП1	×	ИПА	+	↑
ИП3	XY	–	x<0	70	Cx	ИП0	×	XY	1
+	ИП2	×	–	ИПВ	2	0	×	–	В/О

Содержание регистров памяти:

П1 – δ биение детали, мм;

П2 – жесткость пружины K, Н/мм;

П3 – предварительный статический натяг инструмента в деталь Δ_1, мм;

П0 – жесткость материала, умноженная на площадь контакта, $E \cdot s$;

П6 – текущее время процесса t_i, с;

П7 – половина шага интегрирования $h/2$, с;

ПА – x_i – координата вершины инструмента, мм;

ПВ – скорость изменения координаты $\dot{x}_i$, мм/с.

Операторами x < 0, 70, Cx реализована единичная функция. В случае, если содержимое регистра X – значение $\Delta_1 - (x + \delta \sin \omega t)$ – отрицательное, этот регистр очищается; если положительное, то сохраняется и умножение производится на результат этой операции. Переключатель "Р–Г" установлен в положение "Р".

Для оценки точности вычислений, а следовательно, для правильного выбора шага интегрирования воспользуемся кратным изменением шага и по формуле Рунге (53) определим погрешность вычисления производной $\dot{x}$ в узлах (общих точках). Вычислим 24 значения функции $x(t)$ и скорости ее изменения $\dot{x}$ с шагом 0,0025 с, затем 12 значений с шагом в 2 раза большим – 0,005 с, после этого шесть значений с шагом 0,01 с и наконец три значения с шагом 0,02 с. Используя формулу Рунге (53), рассчитаем относительную погрешность удвоения шага.

Предположим, что допускаемая погрешность вычислений 5%. В таком случае необходимо будет ограничиться шагом 0,01 с (табл. 9).

Такой подход к решению дифференциальных уравнений хотя и занимает больше времени на вычисления, например, при нашем подсчете на вычисление ушло $(24 + 12 + 6 + 3) \cdot 1{,}5 = 67{,}5$ мин, т. е. около часа, зато позволяет выбрать такой шаг, который при решении уравнения даст наибольшую информацию. Если необоснованно будет выбран слишком большой шаг, он сведет на нет все вычисления, поскольку погрешность может нарастать; если слишком малый,

Сравнение численного решения дифференциального уравнения (54) при $\omega = 100$ рад/с с различными шагами интегрирования

Таблица 9

Время t, с	$h = 0{,}0025$		$h = 0{,}005$			$h = 0{,}010$			$h = 0{,}02$		
	x	$\dot{x}$	x	$\dot{x}$	$\lvert\delta\rvert$	x	$\dot{x}$	$\lvert\delta\rvert$	x	$\dot{x}$	$\lvert\delta\rvert$
0,0000	0	0	0	0	0	0	0	0	0	0	0
0,0025	−1,2952858−04	−1,4451787−01									
0,0050	−7,9422813−04	−3,8568747−01	−8,225508 −04	−4,062312 −01	1,37−03						
0,0075	−2,0501908−03	−6,1751944−01									
0,0100	−3,8739417−03	−8,3988484−01	−3,9999645−03	−8,5843596−01	1,24−03	−3,2425758−03	−7,9062163−01	4,52−03			
0,0125	−6,2415178−03	−1,052529									
0,0150	−9,1282062−03	−1,2551138	−9,3423288−03	−1,2718181	1,11−03						
0,0175	−1,2508393−02	−1,4472647									
0,0200	−1,6355531−02	−1,6286184	−1,6648855−02	−1,6436121	1,16−03	−1,5281951−02	−1,5890804	3,63−03	−1,2504883−02	−1,4808749	7,21−03
0,0225	−2,064223 −02	−1,7988699									
0,0250	−2,5340455−02	−1,9352543	−2,5704739−02	−1,9232135	8,03−03						
0,0275	−3,0042891−02	−1,7630489									
0,0300	−3,3835699−02	−1,2095679	−3,4085698−02	−1,1841846	1,69−03	−3,3088657−02	−1,365423	1,21−02			
0,0325	−3,5797993−02	−3,0736769−01									
0,0350	−3,5139373−02	8,7247125−01	−3,5230723−02	9,0265024−01	2,01−03						
0,0375	−3,1289191−02	2,2265034									
0,0400	−2,3966722−02	3,679291	−2,3904863−02	−3,6545904	1,78−03	−2,4321587−02	3,4586436	1,31−02	−1,3181006−02	2,4282297	6,87−02
0,0425	−1,3226009−02	4,9376991									
0,0450	5,2872185−04	6,0167928	6,9852619−04	6,034173	1,16−03						
0,0475	1,655876 −02	6,7386657									
0,0500	3,3836882−02	7,0008491	3,4049791−02	7,0069336	4,06−04	3,2076605−02	6,9354306	4,77−03			
0,0525	5,1120258−02	6,7347575									
0,0550	6,7249428−02	6,1647874	6,7239218−02	6,0839498	5,39−04						
0,0575	8,1962741−02	5,6094668									
0,0600	9,5314297−02	5,0751971	9,4919597−02	5,0021524	4,87−03	9,4098442−02	5,0950914	6,20−03	6,8922061−02	3,9670681	7,52−02

то приведет к большому числу промежуточных вычислений, которые не прибавляют никакой информации о графике функции $x = x(t)$.

Для значений $\omega = 100$ рад/с; $K = 1000$ Н/мм; $C = 200$ Н/(мм/с); $\delta = 0{,}05$ мм; $\Delta_1 = 0{,}01$ мм; $Es = 100$ кН/мм; $\Delta = 1$ мм результаты расчета приведены в табл. 10. Графическое решение на основе этих данных показано на рис. 20.

Для значения $\omega = 20$ рад/с и таких же остальных параметров уравнения результаты расчета приведены в табл. 11, а график показан на рис. 21.

Среднее время счета машины в автоматическом режиме — 1,5 мин, запись результатов 20–30 с.

На построение графика численного решения дифференциального уравнения (54) по 100 точкам (шагам) необходимо около 2,5–3 ч, а с учетом времени выбора шага около 4 ч. Построение графика при этом можно выполнять в процессе автоматического счета микрокалькулятора.

Для решения такого вида задач может оказаться удобным наносить график функции $x(t)$ на уже построенный в том же масштабе времени и перемещений график $x_1(t) = \Delta_1 - \delta \sin \omega t$, показывающий положение кромки детали, вращающейся с биением.

В процессе вычислений по адресу 66 программы 36/34 автоматически вычисляется значение натяга инструмента в динамике ("+" — натяг, "—" — зазор). Поэтому интересно исследовать динамику изменения натяга. Для этого необходимо запрограммировать останов (команда С/П), а все остальные операторы передвинуть на один шаг. Естественно, что и адрес безусловного перехода нужно будет изменить. Кривая изменения натяга, построенная на двух предыдущих графиках в том же масштабе времени, может дать интересные сведения о динамике процесса.

Анализ изменения натяга в процессе с угловой скоростью $\omega = 100$ рад/с показывает невозможность работы на таких частотах вращения деталей, поскольку контакт инструмента и детали не постоянный, а периодически повторяется.

При угловой скорости $\omega = 20$ рад/с натяг существует постоянно, однако максимальное значение натяга смещено по фазе от максимального приближения поверхности детали к инструменту несколько больше, чем на половину периода, амплитуда колебаний натяга составляет ± 0,0025 мм. Это вызвано инерционностью системы установки инструмента. Поэтому максимальный натяг будет в зоне наибольшего удаления детали от вершины инструмента, и, наоборот, минимальный натяг — в зоне наибольшего приближения.

При работе такая система будет не уменьшать биение детали, а возможно

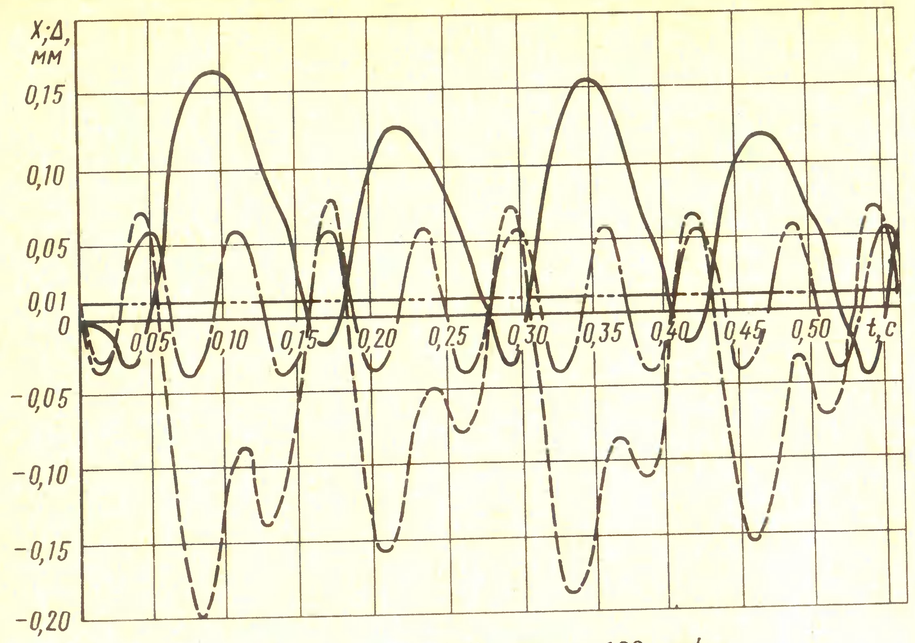


Рис. 20. Графическое решение уравнения (54) при $\omega = 100$ рад/с:
——— – траектория вершины инструмента — — – натяг инструмента при вращении детали (динамический); — · — – траектория кромки детали; — ·· — – предварительный натяг (статический)

Таблица 10

Численное решение дифференциального уравнения второго порядка (54) методом Рунге–Кутта, $\omega = 100$ рад/с

Время, с	x, мм	$\dot{x}$, мм/с	Натяг инструмента Δ, мм
0,00	0,	0,	1,0 –02
0,01	–3,2425758–03	–7,9062163–01	–2,8830976–02
0,02	–1,5281951–02	–1,5890804	–2,0182922–02
0,03	–3,3088657–02	–1,365423	3,6032652–02
0,04	–2,4321589–02	3,4586436	7,2161717–02
0,05	3,2076605	6,9354306	2,5869608–02
0,06	9,4098442–02	5,0950914	–7,0127672–02
0,07	1,3506813–01	3,1491722	–1,5791747–01
0,08	1,5805352–01	1,4965356	–1,9752143–01
0,09	1,659747 –01	1,366647 –01	–1,7658062–01
0,10	1,6173623–01	–9,397174 –01	–1,2453517–01
0,11	1,4797782–01	–1,7779161	–8,797831 –02
0,12	1,267333 –01	–2,4489543	–8,990466 –02
0,13	9,935325 –02	–3,0116857	–1,3774166–01
0,14	6,6799416–02	–3,4826295	–1,0632979–01
0,15	3,0072752–02	–3,8416719	–5,2587137–02
0,16	–9,589727 –03	–3,5044859	3,3984906–02
0,17	–1,8938374–02	–2,0400527	7,7008254–02
0,18	2,533605 –02	5,8225993	2,2013304–02
0,19	7,6862747–02	4,1265286	–7,4356624–02
0,20	1,0905492–01	2,3759208	–1,4470219–01
0,21	1,249855 –01	8,745266 –01	–1,5681827–01
0,22	1,2749657–01	–3,255597 –01	–1,1705399–01

Продолжение табл. 10

Время, с	x, мм	$\dot{x}$, мм/с	Натяг инструмента Δ, мм
0,23	1,1935287–01	–1,263529	–6,7279846–02
0,24	1,0290442–01	–1,998434	–4,7798704–02
0,25	7,982337 –02	–2,6003173	–6,3313179–02
0,26	5,1195317–02	–3,1109129	–7,9401134–02
0,27	1,7915288–02	–3,5271971	–5,5826424–02
0,28	–1,894591 –02	–3,5685633	1,5280165–02
0,29	–3,5647121–02	1,0016854	7,0646125–02
0,30	9,816488 –03	6,6669222	5,0175636–02
0,31	7,2881777–02	5,6717716	–3,4477803–02
0,32	1,182036 –01	3,6445722	–1,3606987–01
0,33	1,45767 –01	1,917485	–1,8603459–01
0,34	1,5754226–01	4,8734889–01	–1,7427357–01
0,35	1,564865 –01	–6,5226078–01	–1,2534573–01
0,36	1,4533371–01	–1,5420578	–8,5965659–02
0,37	1,2624398–01	–2,2519589	–8,4216769–02
0,38	1,006776 –01	–2,8451009	–1,0559114–01
0,39	6,9647841–02	–3,3445206	–1,0792295–01
0,40	3,4153396–02	–3,7335554	–6,1519243–02
0,41	–4,5772903–03	–3,6169991	2,2378353–02
0,42	–1,9235355–02	1,2678575	6,7038136–02
0,43	1,9615992–02	5,6280633	3,0908492–02
0,44	7,0742694–02	4,1138705	–5,5485059–02
0,45	1,0287648–01	2,3691845	–1,3567962–01
0,46	1,1884017–01	8,7992296–01	–1,5418418–01
0,47	1,2140607–01	–3,2054078–01	–1,1784584–01
0,48	1,1330727–01	–1,2590903	–6,5133021–02
0,49	9,6918701–02	–1,9899878	–3,9409590–02
0,50	7,3961796–02	–2,5835542	–5,0954062–02
0,51	4,554274 –02	–3,0864903	–6,9130019–02
0,52	1,2522726–02	–3,5007441	–5,1940035–02
0,53	–2,4090269–02	–3,5644168	1,4178881–02
0,54	–4,133166 –02	9,2044849–01	7,1174045–02
0,55	7,0251682–04	6,9005352	5,5471783–02
0,56	7,1881148–02	6,3782254	–2,9037989–02
0,57	1,2462509–01	4,2256293	–1,3674744–01
0,58	1,5743578–01	2,3878005	–1,9736285–01
0,59	1,7340347–01	8,5721768–01	–1,9552622–01
0,60	1,7559122–01	–3,7133198–01	–1,5063052–01
0,61	1,6687000–01	–1,3340146–	–1,0880055–01
0,62	1,4956631–01	–2,1002008	–1,0277262–01
0,63	1,2529147–01	–2,7370391	–1,2376470–01
0,64	9,5149656–01	–3,2745859	–1,3123963–01
0,65	6,0168562–02	–3,7007141	–9,1620233–02
0,66	2,158784 –02	–3,9920908	–1,0392947–02
0,67	–8,5316089–03	–1,1988795	5,5138312–02
0,68	6,8022434–03	3,821766	4,4367905–02
0,69	4,6917881–02	3,4264925	–2,5401221–02
0,70	7,2972447–02	1,8261091	–1,0189995–01
0,71	8,4260941–02	4,7345703–01	–1,2204239–01
0,72	8,3290885–02	–6,2476032–01	–8,6221715–02
0,73	7,2587906–02	–1,4777965	–2,8974368–02
0,74	5,4398595–02	–2,0545172	4,6872568–03

Продолжение табл. 10

Время, с	x, мм	$\dot{x}$, мм/с	Натяг инструмента Δ, мм
0,75	3,3431922–02	–2,2192869	–4,0404997–03
0,76	8,5287143–03	–2,747517	–2,6911737–02
0,77	–2,1234856–02	–3,1890436	–1,8823775–02
0,78	–5,3432328–02	–2,6281424	3,7039281–02
0,79	–5,3748035–02	3,0639868	7,8340306–02
0,80	8,0787835–03	8,5192937	4,930241 –02

Таблица 11

Численное решение дифференциального уравнения второго порядка (54) методом Рунге–Кутта, ω=20 рад/с

Время, с	x, мм	$\dot{x}$, мм/с	Натяг инструмента Δ, мм
0,00	0,	0,	1,0 –02
0,01	–1,5965028–03	–4,289558 –01	1,563036 –03
0,02	–9,8171688–03	–1,2069139	3,46252 –04
0,03	–2,4522782–02	–1,6245631	6,290657 –03
0,04	–3,9654985–02	–1,2998707	1,378718 –02
0,05	–4,8891397–02	–5,327205 –01	1,6817845 –02
0,06	–5,0772749–02	9,0198 –02	1,4170793 –02
0,07	–4,8800362–02	2,2813332–01	9,527875 –03
0,08	–4,7474216–02	1,601664 –02	7,495535 –03
0,09	–4,8305046–02	–1,4184456–01	9,612665 –03
0,10	–4,9101958–02	3,765894 –02	1,3636085–02
0,11	–4,651486 –02	4,9872 –01	1,6090039–02
0,12	–3,9124981–02	9,496132 –01	1,5351922–02
0,13	–2,84214 –02	1,1418893	1,2696332–02
0,14	–1,725355 –02	1,0626242	1,0504144–02
0,15	–7,4494444–03	8,9734551–01	1,0408447–02
0,16	1,0652675–03	8,371006 –01	1,1853443–02
0,17	9,7710288–03	0,1812746–01	1,3006029–02
0,18	1,9518575–02	1,0205089	1,260745 –02
0,19	2,9740579–02	9,973496 –01	1,085232 –02
0,20	3,8849672–02	8,0339779–01	8,990456 –03
0,21	4,5462446–02	5,1704738–01	8,116346 –03
0,22	4,9267422–02	2,5705015–01	8,4482938–03
0,23	5,0895943–02	8,2345518–02	8,839226 –03
0,24	5,1095659–02	–3,9469481–02	8,7640789–03
0,26	4,7226597–02	–3,8490417–01	7,1254085–03
0,27	4,2187916–02	–6,2127793–01	6,6228302–03
0,28	3,489509 –02	–8,2598156–01	6,7710627–03
0,29	2,5937316–02	–9,5124044–01	7,3148042–03
0,30	1,6140697–02	–9,9823533–01	7,8091093–03
0,31	6,1162682–03	–1,0034314	8,0190377–03
0,32	–3,9001656–03	–9,9947566–01	8,0680288–03
0,33	–1,3850576–02	–9,8778525–01	8,2595176–03
0,34	–2,3542144–02	–9,4333361–01	8,7803286–03
0,35	–3,2512172–02	–8,4168090–01	9,5532962–03
0,36	–4,016173 –02	–6,8172831–01	1,0333625–02
0,37	–4,6009539–02	–4,8609469–01	1,0924695–02

Продолжение табл. 11

Время, с	x, мм	$\dot{x}$, мм/с	Натяг инструмента Δ, мм
0,38	–4,9844658–02	–2,8281888–01	1,1313554–02
0,39	–5,1679605–02	–8,6783707–02	1,1629878–02
0,40	–5,1586595–02	1,039692 –01	1,1995862–02
0,41	–4,9578349–02	2,9700831–01	1,241195 –02
0,42	–4,5625527–02	4,9167504–01	1,2766708–02
0,43	–3,9770294–02	6,7481339–01	1,2940044–02
0,44	–3,2221075–02	8,2806465–01	1,2897558–02
0,45	–2,3348057–02	9,3878491–01	1,2700007–02
0,46	–1,3593726–02	1,005103	1,2435993–02
0,47	–3,3783018–03	1,0322051	1,2147731–02
0,48	6,9322131–03	1,0246775	1,181233 –02
0,49	1,6993038–02	9,8204684–01	1,1383398–02
0,50	2,6434981–02	9,0063682–01	1,0846973–02
0,51	3,485921 –02	7,7901109–01	1,0242384–02
0,52	4,1881772–02	6,2176762–01	9,6367431–03
0,53	4,7193413–02	4,3873758–01	9,0820857–03
0,54	5,0592585–02	2,4094159–01	8,5930783–03
0,55	5,1977697–02	3,7045207–02	8,1591623–03
0,56	5,1318736–02	–1,6708265–01	7,7740025–03
0,57	4,8639369–02	–3,6617157–01	7,4540618–03
0,58	4,4022936–02	–5,5327854–01	7,2324232–03
0,59	3,7633067–02	–7,1959276–01	7,1371056–03
0,60	2,9722817–02	–8,5633956–01	7,1743187–03
0,61	2,062282 –02	–9,5704857–01	7,3293481–03
0,62	1,0712258–02	–1,0184433	7,5809805–03
0,63	3,8999107–04	–1,0396351	7,9150988–03
0,64	–9,9420883–02	–1,0207178	8,3265803–03
0,65	–1,9884641–02	–9,6210505–01	8,8104446–03
0,66	–2,9045875–02	–8,6503252–01	9,3520171–03
0,67	–3,7054863–02	–7,3255995–01	9,9249705–03
0,68	–4,358295 –02	–5,700638 –01	1,0498026–02
0,69	–4,836569 –02	–3,8477609–01	1,1043859–02
0,70	–5,1214227–02	–1,8475044–01	1,1543147–02
0,71	–5,2023022–02	2,1992946–02	1,198204 –02
0,72	–5,0763617–02	2,2770951–01	1,2347041–02
0,73	–4,7484858–02	4,247268 –01	1,2622618–02
0,74	–4,2312691–02	6,053081 –01	1,2793567–02
0,75	–3,5449713–02	7,6193976–01	1,2849698–02
0,76	–2,7169849–02	7,6193976–01	1,2849698–02
0,77	–1,7806439–02	9,7823258–01	1,2616366–02
0,78	–7,7357225–03	1,029312	1,234178 –02
0,79	2,6405443–03	1,0394761	1,1976444–02
0,80	1,2910825–02	1,0084671	1,1533078–02

даже увеличивать. Изменяя параметры дифференциального уравнения, а следовательно, и реальной системы, можно изменить относительный сдвиг этих кривых и таким образом непосредственно влиять на изменение формы детали.

Заметим, что погрешности формы детали, например огранка, некруглость, вызывают периодическое воздействие возмущающей силы, но с другой частотой; при этом картина сдвига по фазе натяга от возмущающей силы остается такой же.

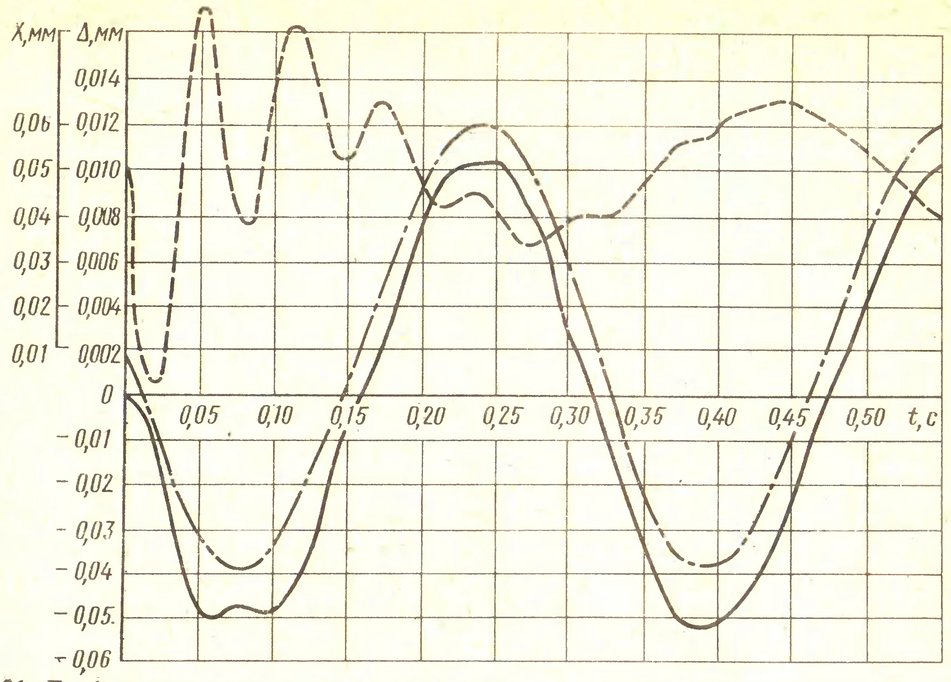


Рис. 21. Графическое решение уравнения (54) при $\omega = 20$ рад/с (обозначения те же, что на рис. 20)

Отсюда следует вывод, что при жестком закреплении детали и упругом одномассовом инструменте периодические погрешности детали уменьшаются мало, а в некоторых случаях даже возрастают.

При использовании гидравлических демпфирующих устройств демпфирующий член может иметь вид $C\dot{x}^2$. В уравнении (54) принято допущение, что площадь пятна контакта не изменяется в процессе изменения натяга. Однако при обработке тороидальным или сферическим инструментом с небольшими радиусами ($r \leqslant 3$ мм) образующих поверхностей и незначительными натягами площадь пятна контакта может изменяться приблизительно пропорционально квадрату или даже кубу натяга:

$$s \approx K_1 (1 + \Delta_1 - x - \delta \sin \omega t)^3 . \qquad (55)$$

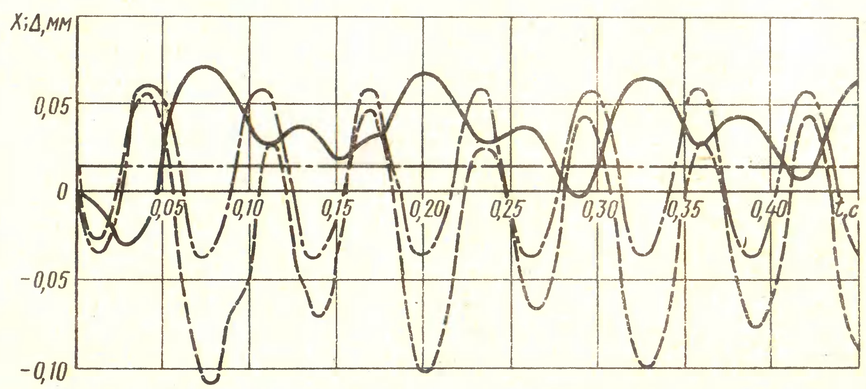


Рис. 22. Графическое решение дифференциального уравнения (54) с нелинейностью (55) при $\omega = 100$ рад/с (обозначения те же, что на рис. 20)

Подстановка этих двух нелинейностей в уравнение (54) существенно усложняет его анализ, но практически не влияет на изменение программы решения этого уравнения.

Программа 37/34. Подпрограмма вычисления значения функции (54) с нелинейным членом (55) и квадратичным демпфированием $C\dot{x}^2$. : До адреса (54) использовать программу 35/34.

ИП6	ИП2	X	sin x	ИП1	X	ИПА	+	П4	ИП2
1/x	XY	–	П3	x<0	71	Cx	ИП2	x^2	X
ИП3	1	+	↑	x^2	X	X	ИП4	1	+
ИП2	X	–	ИПВ	x<0	91	/–/	ИПВ	X	ИП0
X	–	В/О							

Для хранения величины $\omega = 100$ рад/с, $K = 1000$ Н/мм и натяга $\delta = 0{,}01$ мм использован один регистр памяти П2. Его можно применить для получения модуля упругости материала $100^2 = 10\,000$. П0 – коэффициент демпфирования C.

В формуле (55) K_1 принят равным единице. Содержание остальных регистров памяти аналогично программе 36/34. Эти мероприятия позволили один регистр памяти П3 использовать для хранения расчетного динамического натяга и индикации его при считывании показаний x_i и $\dot{x}_i$. За один цикл расчетов сразу получаем три расчетные величины.

Численное решение уравнений (54) с нелинейностью (55) приведено в табл. 12. График функции $x = x(t)$ показан на рис. 22.

Анализ решения уравнения (54) с учетом нелинейностей свидетельствует, что это уравнение более правдоподобно отражает поведение реальной системы, поскольку натяг инструмента не может быть больше нескольких сотых долей миллиметра, поэтому вторая модель в большей степени отражает реальный процесс.

Численное решение уравнения (54) с нелинейностью (55) методом Рунге–Кутта, $\omega = 100$ рад/с *Таблица 12*

Время, с	Координата x, мм	Скорость $\dot{x}$, мм/с	Натяг Δ, мм
0,00	0,	0,	1, –02
0,01	–3,2755543–03	–8,015186–01	–2,6953466–02
0,02	–1,5300704–02	–1,5413402	–2,0540405–02
0,03	–3,1806481–02	–1,1209375	3,3828268–02
0,04	–1,8487822–02	3,7822453	5,6553997–02
0,05	2,8722154–02	4,3513863	4,3533023–02
0,06	5,9784836–02	1,8162431	–3,0511134–02
0,07	7,0736636–02	4,6722406–01	–9,447023 –02
0,08	6,9774913–02	–6,409041 –01	–1,0928825–01
0,09	5,877583 –02	–1,4889927	–6,9783639–02
0,10	4,137292 –02	–1,9258395	–4,712121 –03
0,11	2,8052177–02	– 3,7171726–01	2,6692613–02
0,12	3,3686793–02	9,359518 –01	3,268908 –03
0,13	3,8022226–02	– 1,0060595–01	–4,8625909–02
0,14	3,191842 –02	–1,083643	–7,1553782–02
0,15	1,7500986–02	–1,7298018	–4,0501017–02
0,16	–1,520818 –03	–1,5645325	2,5463762–02
0,17	3,271139 –03	2,616779	4,7595111–02

Продолжение табл. 12

Время, с	Координата x, мм	Скорость $\dot{x}$, мм/с	Натяг Δ, мм
0,18	3,7825973–02	3,2612116	2,0727263–02
0,19	6,0133936–02	1,3281221	–5,7915668–02
0,20	6,7086383–02	1,1305138–01	–1,0319535–01
0,21	6,2764113–02	–9,4466963–01	–9,465987 –02
0,22	4,9385293–02	–1,6561592	–3,9448417–02
0,23	3,0895403–02	–1,6155412	2,0935911–02
0,24	2,754844 –02	8,8028516–01	2,4218774–02
0,25	3,6751933–02	4,767982 –01	–1,5317918–02
0,26	3,621022 –02	–5,738348 –01	–6,436613 –02
0,27	2,5907501–02	–1,4271537	–6,4041365–02
0,28	8,949673 –03	–1,8985818	–1,3011359–02
0,29	–4,3418836–03	–6,6872333–02	4,2369916–02
0,30	1,7399784–02	3,8065763	3,6982393–02
0,31	5,0186083–02	2,1359802	–9,098694 –03
0,32	6,359598 –02	6,7380756–01	–8,2435284–02
0,33	6,4658273–02	–4,4377753–01	–1,04767 –01
0,34	5,533755 –02	–1,3575749–	–7,2107793–02
0,35	3,8892956–02	–1,8615722	–8,033901 –03
0,36	2,5592124–02	–3,7489776–01	2,9062011–02
0,37	3,335443 –02	1,3761557	8,172544 –03
0,38	4,2282526–02	3,106483 –01	–4,5485199–02
0,39	4,0034363–02	–7,4318246–01	–7,8231885–02
0,40	2,8294906–02	–1,5373915	–5,595115 –02
0,41	1,0584498–02	–1,8259237	6,833896 –03
0,42	5,082696 –03	1,2530942	4,3756144–02
0,43	3,2263495–02	3,29824	2,3531299–02
0,44	5,608516 –02	1,4073382	–4,3504187–02
0,45	6,3739543–02	1,800543 –01	–9,680093 –02
0,46	6,0066023–02	–8,8617306–01	–4,3808883–02
0,47	4,7144163–02	–1,6240366	–9,519827 –02
0,48	2,8868409–02	–1,6378203	1,9051681–02
0,49	2,5659491–02	1,0207959	2,7885017–02
0,50	3,8420333–02	9,547848 –01	–1,0214818–02

Последующий анализ экспериментальных данных и уточнение уравнения (54) позволят создать модель, адекватную реальному процессу. Однако и эти графики могут помочь уточнить модель, определить уравнение периодического решения, а следовательно, спектр частот, приведенную массу системы и т. д.

2. ПОСТРОЕНИЕ ФАЗОВЫХ ТРАЕКТОРИЙ

Анализ нелинейных дифференциальных уравнений связан со значительными трудностями, так как в большинстве случаев решение нелинейного дифференциального уравнения не может быть дано в общем виде, в связи с чем приходится прибегать к частным решениям, выполняемым с той или иной степенью приближения. Между тем можно просто получить ответы на некоторые вопросы (например, об устойчивости, наличии автоколебаний, влияния различных коэффициентов, поведении системы в каком-либо частном режиме), используя приближенные методы.

Одним из таких методов является построение фазовых траекторий. Метод фазовых траекторий основан на введении некоторых наглядных понятий и геометрических (точнее сказать, кинематических) представлений. Если объект описывается дифференциальным уравнением порядка n, то состояние этого объекта может быть охарактеризовано n числами, которые можно рассматривать как некоторые векторы n-мерного пространства. Одной точке пространства будет поставлено в соответствие одно определенное состояние объекта (определенная фаза), поэтому такое пространство называют фазовым пространством.

Наибольшее распространение метод фазового пространства нашел при исследовании дифференциальных уравнений второго порядка, поскольку при этом пространство превращается в плоскость. Результаты исследования этим методом позволяют получить все решения дифференциального уравнения для широкой совокупности начальных условий и внешних возмущений в виде прямоугольного или нарастающего сигнала. Метод фазовых траекторий применим для нелинейности любого типа или сочетания нелинейностей. С помощью фазовой траектории можно построить кривую переходного процесса.

Теория фазовых пространств и виды фазовых траекторий описаны в литературе [2, 13, 22], поэтому здесь будут приведены только те преобразования, которые необходимо провести над дифференциальными уравнениями, и методы построения фазовых траекторий и кривых переходных процессов. Анализ характера фазовых траекторий выходит за рамки настоящей книги.

Запишем уравнение (45) в виде

$$\ddot{x} = A\dot{x} + Bx + C. \tag{56}$$

При этом необходимо учесть, что коэффициенты A, B и C уравнения не обязательно являются постоянными (линейное уравнение), они могут быть функциями времени (линейное уравнение с переменными коэффициентами) или функциями самих переменных (нелинейное уравнение). Фазовую траекторию строят в координатах x, $\dot{x}$. Совокупность фазовых траекторий для всех возможных начальных условий, представляющих интерес для анализа, называют фазовым портретом.

Обычно обозначают $\dot{x} = y$, тогда уравнение (56) можно преобразовать:

$$\frac{dy}{dt} = Ay + Bx + C. \tag{57}$$

Если в уравнение (57) время t в явном виде не входит (автономное уравнение), то двух координат – плоскости – для построения траектории достаточно. Поэтому из уравнения (57) исключают время, разделив его на

$$dx/dt = y; \tag{58}$$

получим

$$dy/dx = A + Bx/y + C. \tag{59}$$

Уравнение (59) определяет наклон фазовой траектории

$$y = y(x). \tag{60}$$

4 Зак. 546

Поэтому для построения траектории необходимо разделить переменные в уравнении (59)

$$F(y)dy = F_1(x)dx \tag{61}$$

и проинтегрировать его почленно. Однако разделить переменные и проинтегрировать уравнение (59) аналитически не всегда удается. В этом случае можно достаточно просто проинтегрировать уравнение численными методами.

Рассмотрим один из них.

Для заданных начальных условий x_0 и y_0 определяем угловой коэффициент по формуле (59). Одной из переменных, например x, дают приращение с шагом Δx, тогда значение приращения второй вычисляют по формуле

$$y_i = y_{i-1} + \Delta x\,\Delta y/\Delta x \approx y_{i-1} + \Delta x dy/dx. \tag{62}$$

При положении фазовой траектории, близкой к вертикали, формула (62) неудобна из-за большого значения углового коэффициента фазовой траектории. В этом случае можно рекомендовать приращение по координате Y, а приращение x рассчитывать по формуле

$$x_i = x_{i-1} + \Delta y\,\frac{\Delta x}{\Delta y} \approx x_{i-1} + \Delta y/\frac{dy}{dx}. \tag{63}$$

Условие перехода при интегрировании уравнения (59) от формулы (62) к формуле (63) можно задать при наклоне фазовой траектории $\pm 45^\circ$. Другими словами, если угловой коэффициент фазовой траектории, рассчитанный по формуле (59), составляет

$$dy/dx \geqslant |1|, \tag{64}$$

необходимо использовать формулу (62), в противном случае нужно применять формулу (63). Для того чтобы построить фазовую траекторию во всех положительных и отрицательных квадрантах, необходимо при переходе фазовой траектории через направление -45° изменить знак приращения на противоположный.

Описанный алгоритм может быть запрограммирован для вычисления на микрокалькуляторе. Отметим, что формулы (62) и (63) сходны по внешнему виду: в особенности если деление в формуле (63) заменить умножением на обратную величину углового коэффициента. В этом случае можно использовать косвенное обращение к регистрам памяти, содержащим x и y, и при выполнении условия (64) изменить номера регистров косвенного обращения. Здесь также можно осуществить косвенный переход на подпрограмму с тем, чтобы по одной команде обращаться к подпрограмме вычисления углового коэффициента либо его обратной величины.

Программа 38/34. Построение фазовой траектории дифференциального уравнения второго порядка:

С/П	КПП9	↑	ИПD	×	КИП8	+	КП8	С/П	XY
x^2	1	–	x⩾0	01	ИП8	ИП7	П8	XY	П7
7	3	ИП9	–	П9	3	5	–	x≠0	01
ИПD	/–/	ПD	БП	00	ПП	42	В/О	ПП	42
1/x	В/О	КИП7	ИПD	+	КП7	С/П	...	...	В/О

Вместо многоточия вводят подпрограмму вычисления правой части функции (59).

Содержание регистров памяти:

ПА – начальное условие x_0, в процессе расчета x_i;

П1 – начальное условие y_0, в процессе расчета y_i;

П7 – регистр косвенного обращения к "независимой" переменной, приращение которой постоянно Δ;

П8 – регистр косвенного обращения к "зависимой" переменной, приращение которой вычисляется по формулам (62) или (63);

П9 – регистр косвенного обращения к подпрограмме вычисления dy/dx – 35 записать в П9; $1/(dy/dx)$ – 38 записать в П9.

ПD – постоянное значение приращения Δ.

Если необходимо начинать вычисления при $x_0 = 0$, а $y_0 \neq 0$, в регистры косвенного обращения записывают :1 в П8; 10 в П7; 35 в П9; 0 в ПА; y_0 в П1; Δ в ПD. Если вычисления начинают с оси абсцисс (начальные условия задаются как максимальное перемещение, скорость равна нулю), то в регистры памяти необходимо записать: 1 в П7; 10 в П8; 38 в П9 и 0 в П1, Δ в ПD.

Остальные регистры памяти можно использовать для коэффициентов уравнения (59). Анализ формулы (59) показывает, что значение скорости y_i стоит в знаменателе, следовательно, при $y_i = 0$ формула (59) становится неопределенной. В этом случае полезным оказывается исключить ошибочную ситуацию (индикацию "ЕГГОГ") введением в подпрограмму после вызова числа y в качестве делителя следующего фрагмента:

ИП1 x≠0 А ВП 9 /–/ ÷ ... В/О,

где А – номер оператора деления. Тем самым деление на ноль, как некорректную операцию, заменим делением на 10^{-9}, что на точности вычисления не сказывается.

В программе 38/34 введено три программируемых останова: первый (по адресу 46) индицирует значение независимой переменной, второй (по адресу 08) индицирует значение переменной, рассчитанной по формулам (62) или (63), и наконец третий (по адресу 00) показывает, что зависимая и независимая переменные поменялись местами. В этом месте необходимо поменять порядок вписывания значений в таблицу. Допустим, если первоначально вписывалась величина x_i, а затем y_i, то после индикации значение $-\Delta$, следует при первой индикации теперь заполнять столбец y, а при второй индикации столбец x. При новой индикации Δ необходимо снова изменить порядок записи величин в таблицу.

Программа 38/34 работает быстро, но весьма приближенно, поскольку участок фазовой траектории аппроксимируется прямой с угловым коэффициентом фазовой траектории в начале участка аппроксимации. Можно существенно повысить точность построения фазовых траекторий: дважды вычисляя угловой коэффициент наклона и усредняя его, получать новые значения точек:

$$y^* = y_{i-1} + \Delta x f(x_{i-1}; y_{i-1}); \tag{65}$$

$$y_i = y_{i-1} + \frac{\Delta x}{2}\,[f(x_{i-1}; y_{i-1}) + f(x_i; y^*)].$$

Однако такой алгоритм требует двойного вычисления углового коэффициента касательной на каждом шаге, поэтому производительность его несколько ниже.

Условия изменения направления приращения и выбора зависимых и независимых переменных остаются теми же. Кроме того, в этой программе используется дополнительно два регистра памяти (П2, П0) в качестве буферных.

Программа 39/34. Уточненное построение фазовых траекторий для дифференциальных уравнений второго порядка :

С/П	ПП	62	П2	ИПD	×	КИП8	П0	+	КП8
ПП	57	ИП2	+	2	÷	↑	ИПD	×	ИП0
+	КП8	С/П	XY	x^2	1	–	x⩾0	01	ИП8
ИП7	П8	XY	П7	1	0	3	ИП9	–	П9
5	0	–	x≠0	01	ИПD	/–/	ПD	БП	00
ПП	64	В/О	ПП	64	1/x	В/О	КИП7	ИПD	+
КП7	С/П	КПП9	В/О	...	...	...	В/О		

Все сказанное в отношении программы 38/34 остается в силе и для этой программы, кроме того что в П9 записывают число 50 при $x_0 = 0$ или 53 при $y_0 = 0$.

Иногда оказывается удобным индицировать значения x и y не после каждого цикла итераций, а через несколько шагов. В этом случае точность вычислений остается достаточно высокой, поскольку шаг интегрирования невелик, а график можно строить и через большие промежутки. С этой целью в программу введен оператор цикла, и при выходе из цикла переменные размещаются в операционных регистрах : x_i в регистре X, а y_i в регистре Y. Это значительно упрощает работу с программой.

Программа 40/34. Построение фазовых траекторий с выборочной индикацией значений :

ПП	69	П2	ИПD	×	КИП8	П3	+	КП8	ПП
65	ИП2	+	2	÷	↑	ИПD	×	ИП3	+
КП8	L0	31	XY	П3	ИП1	ИПА	С/П	П0	ИП3
↑	XY	x^2	1	–	x⩾0	00	ИП8	ИП7	П8
XY	П7	1	1	9	ИП9	–	П9	5	8
–	x≠0	00	ИПD	/–/	ПD	БП	00	ПП	71
В/О	ПП	71	1/x	В/О	КИП7	ИПD	+	КП7	КПП9
В/О	...	...	В/О						

Содержание регистров памяти такое же, как и в предыдущих программах, но первоначально в П0 заносят 1, в П9 – 58 для $x_0 = 0$ или 61 для $y_0 = 0$. При индикации результата и записи полученных значений в таблицу результатов в регистр X вводят требуемое число циклов повторений без индикации и нажимают клавишу С/П. Если необходимо индицировать результат вычисления после каждого шага, вместо команды П0 (адрес 28) вводят команду Кноп (нет операции).

В качестве примера рассмотрим порядок построения фазовой траектории уравнения Ван дер Поля [9] :

$$\ddot{x} = -x + \mu(1 - x^2)\dot{x}. \qquad (66)$$

Вводя подстановку $\dot{x} = y = dx/dt$, получим

$$dy/dx = \dot{y} = -x + \mu(1 - x^2)y.$$

Исключив t, можем записать

$$\frac{dy}{dt}\Big/\frac{dx}{dt} = -\frac{x}{y} + \mu(1 - x^2)\frac{y}{y},$$

откуда $dy/dx = \mu(1 - x^2) - x/y$. (67)

Программа 41/34. Подпрограмма вычисления углового коэффициента уравнения Ван дер Поля (66) : До 71-го адреса ввести программу 40/34.

ИП4	1	ИПА	x^2	–	×	ИПА	ИП1	x=0	84
ВП	6	/–/	÷	–	В/О				

Содержание регистров памяти такое же, как и в предыдущей программе. В регистр П4 введено значение μ.

Индикация микрокалькулятора при расчете координат фазовой траектории показана в табл. 13.

Первоначально в табл. 13 заносились координаты через каждый цикл расчетов, затем с достаточной для построения точностью записывались значения координат через 4–5 циклов. Время расчета одного цикла составляет около 20 с.

Таблица 13

Координаты точек фазовой траектории уравнения Ван дер Поля (66) при $\mu = 0{,}4$, $y_0 = 0$, $x_0 = 0{,}5$, $x_0 = 3$

x	y	x	y
5, –01	0,	3,	0,
4,6428594 –01	–2, –01	2,9915255	–2, –01
3,5003948–01	–4, –01	2,9597641	–4, –01
1,2512101 –01	–6, –01	2,882738	–6, –01
-7,487899 –02	–6,8960972–01	2,7025211	–8, –01
-8,7487899–01	–2,154311 –02	1,7535331	–1,3038477
-7,0576526 –01	5,8450689 –01	9,535331 –01	–1,7478995
1,1463419 –01	1,1749419	–1,0464669	–2,2796142
9,1463419–01	1,0388849	–1,8457253	–1,272171
1,3477571	2,204276 –01	–2,1066934	–2,72171 –01
1,2237057	–5,795824 –01	–1,9301758	7,27829 –01
4,9748028–01	–1,3264992	–1,1507828	1,421043
-3,0251972–01	–1,6848115	–1,507828 –01	2,0132768
-1,1025197	–1,4845103	8,492172 –01	2,1701377
-1,6109816	–6,74987 –01	1,6970863	1,3365583
-1,7205607	1,25013 –01	2,0042883	3,365583 –01
-1,3708026	9,25013 –01	1,8735357	–6,634417 –01
-5,701057 –01	1,5710288	8,951146 –01	–1,5438859
2,298943 –01	1,9465532	–1,048854 –01	–2,0771277
1,0298943	1,8631715	–1,1048854	–2,0100065
1,6580562	1,0871223	–1,8095745	–1,0083721
1,8863525	2,871223 –01	–2,0055866	–8,3721 –03
1,5682003	–9,128777 –01	–1,5938632	9,916279 –01
9,658452 –01	–1,4241666	–5,668825 –01	1,7386909
1,658425 –01	–1,9045648	4,331175 –01	2,1447134
-6,341548 –01	–2,0990885	1,4331175	1,6913151

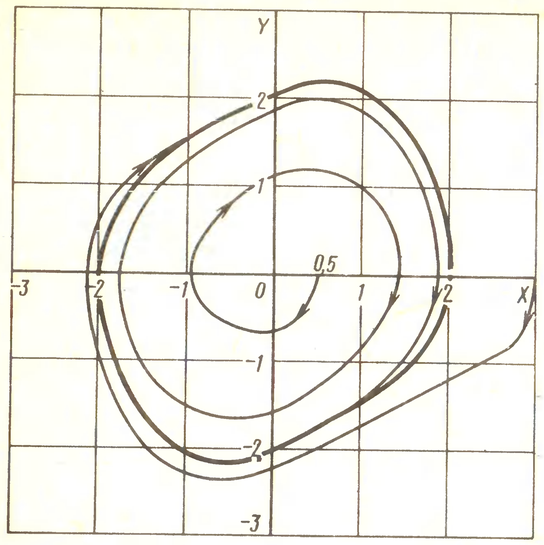


Рис. 23. Фазовый портрет уравнения Ван дер Поля при $\mu = 0{,}4$, начальные условия $y_0 = 0$; $x_{01} = 0{,}5$; $x_{02} = 3{,}0$

Исходные данные для примера: $\mu = 0{,}4$; $y_0 = 0$; $x_{01} = 0{,}5$; $x_{02} = 3{,}0$.

Анализ фазовых траекторий (рис. 23) показывает, что система описываемая уравнением (66), имеет устойчивое положение равновесия, представляющее собой предельный цикл, и при любом отклонении начальных условий она с течением времени выходит на этот цикл. Если начальные условия находятся внутри цикла, то фазовая траектория представляет собой спираль, развертывающуюся до цикла, если они находятся вне цикла, то спираль, наворачивающуюся на цикл.

Указанный предельный цикл характеризует наличие в системе устойчивых периодических колебаний, к которым система приходит вне зависимости от начального отклонения.

Характеристики поведения динамических систем в зависимости от их фазовых портретов можно найти в литературе [13, 22].

Фазовую траекторию можно использовать для построения кривой переходного процесса. Очевидно, что

$$\Delta x = \int_{t_1}^{t_2} \dot{x} dt = \int_{t_1}^{t_2} y dt. \qquad (68)$$

По формуле трапеций определенный интеграл (68) можно заменить площадью прямолинейной трапеции:

$$\Delta x \approx \frac{y(t_1) + y(t_2)}{2} \Delta t = \frac{y_{i-1} + y_i}{2} \Delta t. \qquad (69)$$

Величины, входящие в формулу (69), кроме Δt, можно получить из фазовой траектории. Поэтому

$$\Delta t = \frac{2(x_i - x_{i-1})}{y_i + y_{i-1}}. \qquad (70)$$

Для упрощения вычислений формулу (70) можно преобразовать с учетом выражения (62):

$$\Delta t = \frac{2\Delta x}{y_i - \Delta x\, dy/dx + y_i} = \frac{2\Delta x}{2y_i - \Delta x\, dy/dx}, \qquad (71)$$

или еще проще:

$$\frac{1}{\Delta t} = \frac{y_i}{\Delta x} - \frac{dy/dx}{2}. \qquad (72)$$

Очевидно, что значение углового коэффициента можно брать либо полученное из формул (62), (63), либо усредненное, как в формуле (65).

В первом случае программа получается существенно проще, поскольку для хранения значения коэффициента (но не его обратной величины) можно использовать регистр памяти, записав это значение в конце подпрограммы вычисления функции.

Во втором случае задача несколько усложняется, поскольку в зависимости от поведения фазовой траектории вычисляется либо усредненный коэффициент наклона, либо его обратная величина. Легко убедиться, что замена, аналогичная переходу от (62) к (63), к выражению, подобному (72), не приводит. Поэтому значение углового коэффициента, необходимое для вычисления по формуле (72), определяется из среднего значения угла наклона или его обратной величины.

Программа 42/34. Построение переходной кривой процесса, описываемого дифференциальным уравнением (66), с использованием фазовых траекторий:

ИПА	ПС	ПП	80	П2	ИПD	×	КИП8	П0	+
КП8	ПП	76	ИП2	+	2	÷	П2	ИПD	×
ИП0	+	КП8	ИП2	ИП3	ИП9	2	×	–	x<0
34	XY	1/x	XY	XY	ИП1	ИПА	ИПС	–	÷
XY	2	÷	–	С/П	ИП2	x^2	1	–	x⩾0
00	ИП8	ИП7	П8	XY	П7	ИП3	ИП9	–	П9
вх	–	x⩾0	00	ИПD	/–/	ПD	БП	00	ПП
82	В/О	ПП	82	1/x	В/О	КИП7	ИПD	+	КП7
КПП9	В/О	...	...	...	В/О				

Содержание регистров памяти:

П1 – скорость изменения y_i координаты $x(t)$;

П2 – буферный регистр, используемый для хранения коэффициента угла наклона или его обратной величины, кроме того, используется для вычисления среднего значения угла наклона;

П3 – число 141, полученное как сумма адресов косвенного обращения к подпрограмме (141 = 69+72);

П7 – регистр косвенного обращения к независимой переменной, которой сообщают приращение с постоянным шагом;

П8 – регистр косвенного обращения к зависимой переменной, значение которой вычисляют, используя угловой коэффициент фазовой траектории;

П9 – номер 69 или 72 адреса косвенного перехода на подпрограмму. Если в качестве независимой переменной выступает x, то П9 = 69, если y, то П9 = 72; замена адресов производится автоматически в программе;

П0 – буферный регистр памяти;

ПА – координата x_i;

ПС – координата x_{i-1};

ПD – приращение независимой координаты Δ.

Используем эту программу для построения кривой переходного процесса системы, описываемой уравнением (66).

Значение коэффициента $\mu = 0{,}4$. Начальные условия: $x_0 = 3{,}0$; $y_0 = 0$. Фазовый портрет этого процесса приведен на рис. 23. Время расчета одного значе-

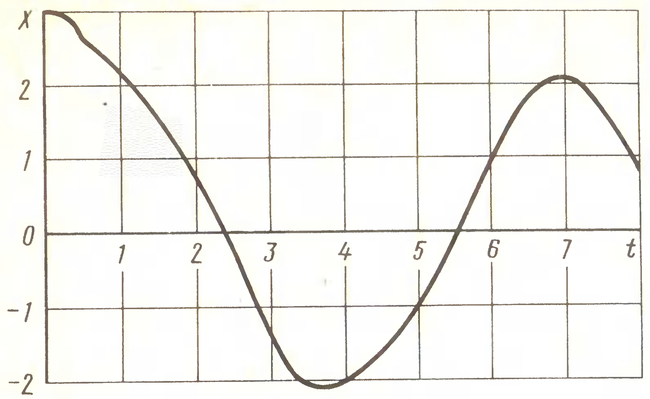


Рис. 24. График переходного процесса уравнения Ван дер Поля при $\mu = 0{,}4$; $y_0 = 0$; $x_0 = 3{,}0$

ния 25 с. Ограниченность программной памяти не позволяет вычислить все в автоматическом режиме. Вычисление t, как обратной величины выражения (72), и расчет абсолютного времени $t_i = t_{i-1} + \Delta t$ выполняем в ручном режиме: клавиши F 1/x; ИПВ; +; ПВ.

Для хранения абсолютного значения времени использован любой свободный регистр памяти (П5, П6, П8). Координаты переходного процесса приведены в табл. 14.

На рис. 24 изображено решение дифференциального уравнения (66), построенное с помощью интегрирования фазовой траектории. Построение графиков переходных процессов автономных систем существенно выгоднее численного решения методом Рунге–Кутта, во-первых, из-за высокой скорости построения так как координаты каждой новой точки вычисляются за 20–25 с, а шаг может быть принят существенно бо́льшим, чем в методе Рунге–Кутта. Во-вторых, одновременно полученная фазовая траектория позволяет оценить поведение системы в целом.

Программируемый микрокалькулятор позволил применить с высокой точностью графический метод, который ранее широко не использовался из-за невозможности точного отсчета координат фазовой траектории при ее интегрировании.

Таблица 14

Координаты переходного процесса уравнения (66) при $\mu = 0{,}4$; $x_0 = 3{,}0$; $y_0 = 0$

Приращение времени Δt	Абсолютное время t	Координата x
0,	0,	3,
1,0587128–01	1,9061752–01	2,959764
2,5745271–01	6,0212242–01	2,702521
1,9023008–01	1,1801168	2,1535331
1,5960527–01	1,513254	1,7535331
1,3660613–01	1,7972941	1,3535331
1,1833811–01	2,0425703	9,535331 –01
1,0413926–01	2,2574611	5,535331 –01
9,3725095–02	2,4496573	1,535331 –01
8,6981746–01	2,626531	–2,464669 –01
8,4144252–02	2,7957113	–6,464669 –01
8,6335752–02	2,9665226	–1,0464669
9,7449246–02	3,1542613	–1,4464669
6,9173136–02	3,3337401	–1,7552189
6,4902217–02	3,4646008	–1,9218018

Приращение времени Δt	Абсолютное время t	Координата x
6,7548384–02	3,5976379	–2,0376269
7,62612 –02	3,7450042	–2,1066934
1,1919363–01	2,9467072	–2,1175822
1,3626097–01	4,1988521	–2,0328841
1,9313916–01	4,5555841	–1,7702896
1,6606784–01	4,9352154	–1,3507828
1,3445384–01	5,2177149	–9,507828 –01
1,1485919–01	5,4562228	–5,507828 –01
1,0186963–01	5,6657693	–2,507828 –01
9,3930943–02	5,702955	2,492172 –01
9,0853595–02	5,8855589	6,492172 –01
9,3926814–02	6,0709312	1,0492172
1,0854026–01	6,2785499	1,4492172
7,642096 –02	6,4393472	1,6970863
7,162511 –02	6,5837059	1,8612692
7,5034617–02	6,731202	1,9696514
8,5224645–02	6,8957675	2,0244489
1,1496262–01	7,0923342	2,0086424

3. ЧИСЛЕННЫЕ МЕТОДЫ РАСЧЕТА ОПРЕДЕЛЕННЫХ ИНТЕГРАЛОВ

При исследовании математических моделей приходится рассчитывать различные определенные интегралы, причем точное вычисление по методу Ньютона–Лейбница [11] не всегда представляется возможным или целесообразным из-за сложности подынтегральной функции. Существует много методов вычисления определенных интегралов [6, 12, 13]. Наиболее приемлемыми на программируемых микрокалькуляторах являются методы трапеций и парабол, которые сочетают достаточную простоту с высокой точностью.

Метод трапеций. Известно, что определенный интеграл геометрически представляет собой площадь фигуры, ограниченной ординатами начальной и конечной точек интегрирования, осью абсцисс и графиком подынтегральной функции. Если график функции описывает некоторую замкнутую в пространстве фигуру, то определенный интеграл по контуру будет представлять собой площадь фигуры, ограниченную этой замкнутой кривой.

Для элементарной трапеции формула уже применялась (69). В общем виде формула трапеций имеет вид

$$I = \int_a^b f(x)dx \approx \frac{h}{2} \sum_{i=1}^{n} (y_i + y_{i-1}), \qquad (73)$$

где $y_i = f(x_i)$; $h = (b-a)/n$.

Если подынтегральная функция задана параметрическими уравнениями $y = f_1(\epsilon)$, $x = f_2(\epsilon)$, то формула трапеций приобретает вид

$$I = \sum_{i=1}^{n} [(x_i - x_{i-1})(y_i + y_{i-1})/2]. \qquad (74)$$

Алгоритм вычисления по формуле (73) достаточно простой. Исходя из пределов интегрирования и числа циклов приближения, определяют шаг интегри-

рования — параметр цикла — и организуют цикл вычисления значения подынтегральной функции, используя каждый раз в цикле формулу (3). Здесь же в цикле проводится вычисление площади элементарной трапеции и суммируются площади элементарных трапеций. Поскольку в цикле нельзя определить текущее значение переменной при $i = 0$, то после выполнения цикла подынтегральная функция при $i = 0$ вычисляется один раз, что сравнимо с x_{max}, и рассчитывается площадь последней трапеции, и, наконец, при получении всей суммы индицируется результат.

Программа 43/34. Вычисление определенных интегралов по формуле трапеций:

С/П	П0	ПD	ИПВ	ИПА	–	XY	÷	ПВ	ИПD
ИП0	–	ПП	22	L0	09	ИПD	ПП	24	ИП2
ИПВ	×	БП	00	ИПВ	×	ИПА	+	ПС	...
ИП1	XY	П1	+	2	÷	ИП2	+	П2	В/О

Содержание регистров памяти:

П0 — счетчик цикла — число циклов интегрирования;

П1 — значение вычисленной функции на предыдущем шаге;

П2 — регистр-сумматор — значение определенного интеграла;

ПА–ПВ — границы интервала интегрирования;

ПС — текущее значение аргумента подынтегральной функции;

ПD — приращение аргумента.

Вместо многоточия записать программу вычисления требуемой функции, учитывая, что аргумент функции находится в операционном регистре X и регистре ПС, а вычисленное значение функции должно находиться также в операционном регистре X. Перед пуском программы ввести в регистр X число циклов интегрирования и дважды нажать клавишу С/П. При последующих пусках клавишу С/П нажать один раз.

Иногда можно использовать программу более короткую, которая каждый раз осуществляет приращение аргумента, без применения параметра цикла [формула (4)].

Программа 44/34. Упрощенное вычисление определенных интегралов по формуле трапеций:

П0	ИПА	ПП	34	П1	ИПВ	ПП	34	ИП1	+
2	÷	П1	ИПВ	ИПА	–	ИП0	÷	ПВ	КИП0
ИПВ	ИПА	+	ПА	ПП	34	ИП1	+	П1	L0
20	ИПВ	×	С/П	...	...	...	...	В/О	

Содержание регистров памяти такое, как и в предыдущей программе, за исключением того, что регистры ПС и ПD свободны.

Для интегрирования функций, заданных таблично, метод трапеций позволяет составить очень экономичную и быстродействующую программу.

Программа 45/34. Интегрирование функций, заданных таблично, по формуле трапеций:

С/П	ИП1	XY	П1	XY	–	С/П	ИП2	XY	П2
+	2	÷	×	+	БП	00			

Эта программа содержит в регистре П1 независимую переменную, а в регистре П2 — функцию от независимой переменной, заданную таблично.

Несложность программы не означает простоты работы с ней, поскольку машина работает пассивно: принимает и обрабатывает все числа, которые вводит оператор. Монотонная работа по вводу чисел утомляет и рассеивает внимание, поэтому может возникнуть ситуация, когда оператор начинает сомневаться, ввел он первую или вторую переменную из таблицы. Для облегчения работы программу можно несколько усложнить. Например, перед вторым оператором С/П ввести операторы ↑ и Сх, благодаря чему микрокалькулятор будет сообщать пользователю, какое число необходимо вводить. Если на индикаторе не нуль, необходимо ввести первую переменную, если нуль, то вторую переменную. Такой подход сохранен при составлении программ работы с большими массивами чисел, которые необходимо ввести в микрокалькулятор.

Из программы 45/34 достаточно просто получить программу для интегрирования функции, заданной параметрически. Программу необходимо дополнить приращением параметра требуемое число раз, а ручной ввод переменных (операторы С/П) заменить отсылкой к подпрограммам вычисления параметрических функций.

Метод парабол. Для вычисления определенных интегралов иногда элементарную трапецию аппроксимируют криволинейной трапецией, у которой вместо графика функций применена парабола. Для вычисления интегралов по методу парабол применяют формулу Симпсона [32]

$$I = \int_a^b f(x)dx = \frac{h}{3}\left[f(a) + f(b) + 4\sum_{i=0}^{k-1} f(a + h(2i+1)) + 2\sum_{i=0}^{k-1}(a+h2i)\right], \quad (75)$$

где $k = n/2$; $h = (b-a)/n$.

Формула Симпсона дает точность на порядок выше формулы трапеций.

Программа вычисления определенных интегралов по формуле Симпсона разработана для микрокалькулятора "Электроника" Б3-21 [32]. Здесь она приведена несколько измененной и переработанной для микрокалькулятора "Электроника" Б3-34.

Программа 46/34. Вычисление определенных интегралов по формуле Симпсона:

ПП	37	П8	ИП3	ИП2	П3	–	ИП0	÷	П2
ПП	37	1	ПП	21	4	ПП	21	2	БП
13	×	ИП8	+	П8	L0	33	ИП8	3	÷
ИП2	×	С/П	ИП3	ИП2	+	П3	...	...	В/О

Содержание регистров памяти:

П0 — счетчик цикла, первоначально в него заносится требуемое число циклов интегрирования;

П2 — первоначально в него заносится нижний предел интегрирования, в процессе вычислений хранится шаг интегрирования;

П3 — первоначально в него заносится верхний предел интегрирования, в процессе вычислений хранится текущее значение x_i;

П8 — регистр-сумматор, в конце вычислений в нем образуется значение определенного интеграла.

Остальные регистры в программе не используются, в них можно размещать коэффициенты подинтегральной функции. Программу вычисления подынтегральной функции следует записать вместо многоточия так, чтобы она заканчивалась выводом значения функции в регистр X.

4. ПРИЛОЖЕНИЕ МЕТОДОВ ВЫЧИСЛЕНИЯ ОПРЕДЕЛЕННЫХ ИНТЕГРАЛОВ ПРИ МОДЕЛИРОВАНИИ ПРОЦЕССА ПОВЕРХНОСТНОГО ПЛАСТИЧЕСКОГО ДЕФОРМИРОВАНИЯ

При поверхностном пластическом деформировании деталей из легких сплавов, когда тороидальный ролик, выполненный из закаленной стали, имеет модуль упругости гораздо выше, чем поверхность обрабатываемой детали, важное значение имеет взаимное положение осей обрабатываемой торцевой поверхности и тороидального ролика. Пятно контакта имеет сложную форму и вызывает проскальзывание в зоне обработки.

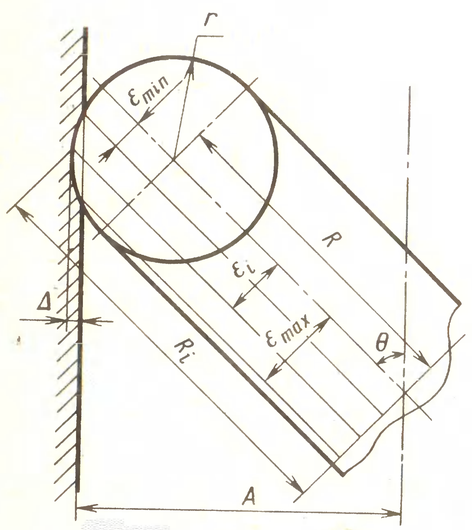
В зависимости от пятна контакта и модуля упругости можно более точно рассчитать необходимые усилия деформирования.

Для упрощения расчетной схемы принимаем, что ролик выполнен идеально тороидальным, из абсолютно жесткого материала. Обрабатываемая деталь идеально плоская и в момент контакта имеет абсолютную податливость. Зазоры в системе СПИД – нулевые. Площадь контакта рассчитаем в проекции на плоскость, параллельную обрабатываемой поверхности и проходящую через зону обработки (рис. 25).

Для определения координат профиля выведены расчетные формулы [33]

$$\epsilon_{max} = r\cos\left(\arccos\frac{r-\Delta}{r} - \theta\right); \qquad (76)$$

$$\epsilon_{min} = r\cos\left(\arccos\frac{r-\Delta}{r} + \theta\right); \qquad (77)$$



$$\epsilon_i = \frac{\epsilon_{max} - \epsilon_{min}}{n} i + \epsilon_{min}; \qquad (78)$$

$$\varphi = \arcsin\frac{R\sin\theta + r - \Delta - \epsilon_i\cos\theta}{R_i\cos\theta}; \qquad (79)$$

$$R_i = R + \sqrt{r^2 - \epsilon_i^2}; \qquad (80)$$

$$\begin{cases} x_i = R_i\cos\varphi; \\ y_i = R_i\sin\varphi\cos\theta - \epsilon_i\sin\theta, \end{cases} \qquad (81)$$

Рис. 25. К расчету площади пятна контакта при ППД плоскости тороидальным роликом

где ϵ_{max}, ϵ_{min} – параметры, характеризующие расстояние от оси симметрии тора до его крайних сечений, контактирующих с плоскостью, мм; r – радиус тора, мм; Δ – натяг, мм; θ – угол наклона торового ролика к плоскости обработки, °; n – число шагов интегрирования; R – радиус от центра вращения до оси тора, мм; φ – пересчетный параметр, определяющий угловое расстояние от оси Y до места контакта торового круга в сечении ϵ_i с плоскостью обработки, °; R_i – значение радиуса окружности при пересечении тора плоскостью на расстоянии ϵ_i от оси симметрии, мм.

Площадь пятна контакта описывается системой параметрических уравнений (81), в которых параметром является расстояние от оси симметрии тора до секущей плоскости. По существу задача определения уравнения линии пересечения тора и плоскости аналитически решена средствами начертательной геометрии. Кроме того, в уравнения входит пересчетный параметр φ (79). Формулы (76) и (77) определяют крайние значения параметра, вычисляются один раз, поэтому программировать их нет смысла.

Для вычисления площади, описываемой уравнениями (81), используем метод трапеций.

Программа 47/34. Расчет площади пятна контакта тороидального ролика и плоскости при поверхностном пластическом деформировании:

ИП5	ИП1	–	ИП5	÷	ИПВ	ИПА	–	×	ИПА
+	ПП	21	L1	00	ИПВ	ПП	21	2	×
С/П	ПС	x^2	ИП6	x^2	–	$\sqrt{x}$	ИП2	+	П7
ИП2	ИП3	sin x	×	ИП6	ИПD	–	+	ИПС	ИП3
cos x	×	–	вх	ИПС	÷	ИП7	×	÷	П4
ИП7	ИП3	cos x	×	×	ИПС	ИП3	sin x	×	–
ИП9	XY	П9	+	2	÷	ИП0	1	ИП4	x^2
–	x<0	74	Cx	$\sqrt{x}$	ИП7	×	П0	–	×
ИП8	+	П8	В/О						

Содержание регистров памяти:

П0 – значение x_{i-1} при первоначальном вводе $x_0 = 0$;

П1 – число шагов и порядковый номер выполняемого шага i (необходимо учитывать, что порядковый номер идет от номера n до 1);

П2 – радиус инструмента R, мм;

П3 – угол наклона инструмента к плоскости обработки θ, °;

П4 – значение синуса пересчетного параметра φ (первоначально не вводится);

П5 – максимальное число шагов интегрирования n в начале счета П1 = П5;

П6 – радиус малого торового круга инструмента r, мм;

П7 – при первоначальном вводе не заполняется;

П8 – текущее значение суммы площадей элементарных трапеций, в начале счета необходимо очистить регистр-накопитель суммы;

П9 – значение y_{i-1};

ПА – значение параметра ϵ_{max}, вычисленное по формуле (76);

ПВ – значение параметра ϵ_{min}, вычисленное по формуле (77);

ПС – текущее значение параметра ϵ_i, при первоначальном вводе не заполняется;

ПD – натяг Δ, мм.

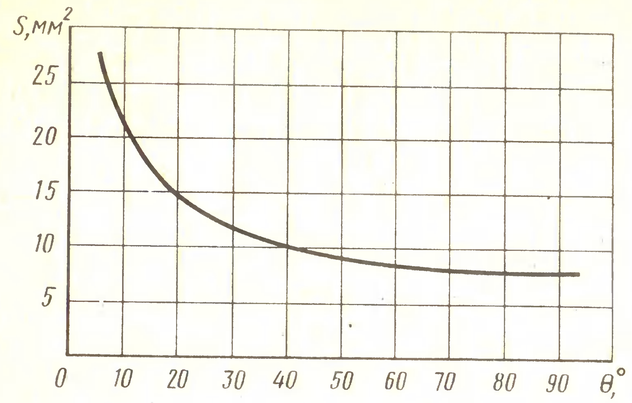


Рис. 26. Зависимость площади пятна контакта от угла установки ролика при одном и том же натяге

В процессе определения пересчетного параметра φ замечено, что нет необходимости вычислять арксинус, а затем синус и косинус. В связи с этим пересчетный параметр хранится в виде синуса, косинус получается с помощью формулы преобразования $\cos x = \sqrt{1 - \sin^2 x}$. Известно, что округления при вычислениях тригонометрических функций могут привести к тому, что значения синуса пересчетного параметра окажутся больше единицы на величину, например, 10^{-5}. С этой целью в программу введен анализ округления. По адресам 71–73 проверяется содержимое подкоренного выражения. Если оно меньше нуля, то содержимое операционного регистра X заменяется нулем, и программа следует дальше; если больше или равно нулю, то непосредственно вычисляет квадратный корень. Такие средства программного анализа ошибок вычислений можно использовать всегда, когда есть риск получить ожидаемое значение вычисляемого операнда, сравнимое с машинным нулем, но не равное нулю.

Среднее время вычисления площади при числе шагов $n = 20$ составляет около 12 мин, при числе шагов $n = 1000$ время вычисления более 9 ч. Необходимо отметить, что точность вычислений микрокалькулятора выражается седьмой значащей цифрой, поэтому ошибка после 1000 вычислений может составлять вторую-третью значащую цифру и такую кратность вычислений выбирать не следует. Следует помнить, что при изменении натяга или угла установки изменяются и пределы параметра ϵ. В этом случае значения ϵ_{min} и ϵ_{max} определяются по формулам (77), (78).

Зависимость площади пятна контакта от угла установки ролика, рассчитанная по разработанной программе, приведена на рис. 26. Использование этой зависимости позволит ориентировочно определить площадь пятна, а следовательно, при известных усилиях и давление в зоне контакта.

5. ИССЛЕДОВАНИЕ ПРОЦЕССА ВИХРЕВОГО ФРЕЗЕРОВАНИЯ РЕЗЬБ

В процессах фрезерования резьб и их вихревого нарезания необходимы быстрое вращение инструмента для получения нужной скорости резания и относительно медленное вращение заготовки – движение круговой подачи. Известные методы требуют перемещения инструмента за один оборот заготовки на один шаг, что не позволяет применять их при совмещенной обработке с точением, сверлением и другими операциями на токарных автоматах.

Был предложен новый способ фрезерования наружных резьб, заключающийся в том, что между двумя последующими касаниями инструмента заготовка совершает один оборот с суммой или разностью части оборота, определяющего значение круговой подачи на зуб фрезы (А.с. 795780 СССР, МКИ³ B23G 1/32).

При работе по этому способу инструмент вращается с частотой, сравнимой с частотой вращения заготовки, определяемой потребностью режимов резания при точении, сверлении или других операциях технологического цикла обработки детали на автомате. Переключения частоты вращения шпинделя не требуется. При этом создаются предпосылки для совмещенной обработки резьбовых поверхностей с другими поверхностями за один проход, следовательно, уменьшается технологическое время на обработку детали.

Однако этот способ накладывает ограничения на кинематику процесса, поскольку здесь взаимосвязаны частоты вращения заготовки, инструмента и осевая подача заготовки на один оборот. С другой стороны, при определенных условиях соотношения частот вращения и диаметров могут стать такими, что мгновенный центр скоростей системы заготовка–инструмент будет точно в зоне резания, поэтому процесс станет невозможным.

Используя разное число зубьев фрезы и различный диаметр ее, можно сделать так, что окружная скорость фрезы будет либо меньше, либо больше окружной скорости заготовки в зоне резания. При этом одинаковое направление линейной скорости может потребовать вхождения инструмента в зону резания задней поверхностью. Такое точение с вхождением инструмента в зону резания задней поверхностью было предложено Г.А. Шаумяном [43] и названо попутным точением. В этом процессе задняя поверхность инструмента отходит от обработанной поверхности, поэтому создаются благоприятные условия для обработки, возможность работы инструмента с большой степенью затупления, благоприятное изменение кинематических углов резания, кинематическое стружкодробление, остывание режущей кромки.

Процесс вихревого фрезерования резьбы (А.с. 795780 СССР, МКИ³ B23G 1/32) отличается от известных еще и тем, что скорость резания в нем зависит не только от окружной скорости инструмента, но складывается векторно с линейной скоростью точки контакта инструмента и заготовки. При этом получается изменение результирующей скорости резания как на входе-выходе инструмента в зону обработки, так и в момент прохождения линии, соединяющей оси.

В обычных условиях скорость резания и скорость подачи отличаются на два порядка, поэтому скорость резания принимают независимой от вектора скорости подачи.

Полное аналитическое выражение изменения скорости резания описано в работах [28, 29], здесь приведем только расчетную схему (рис. 27) и полученные формулы:

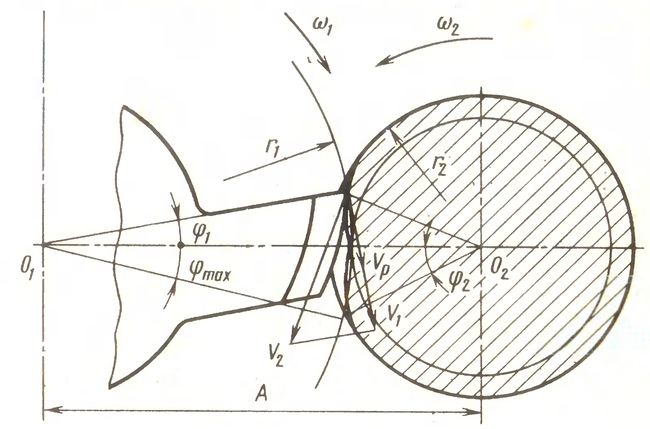


Рис. 27. Схема процесса вихревого фрезерования резьбы

$$V_{рез} = V_1 - V_2\psi; \tag{82}$$

$$V_1 = \frac{\pi d_1 n_1}{1000}; \tag{83}$$

$$V_2 = \frac{\pi d_2 n_2}{1000}; \tag{84}$$

$$\psi = \frac{A\cos\varphi - d_1/2}{A - d_1/2}, \tag{85}$$

где d_1, d_2 – диаметры инструмента и готовой детали соответственно, мм; n_1, n_2 – частота вращения инструмента и заготовки соответственно, об/мин; A – межосевое расстояние, мм; φ – текущий угол поворота инструмента – аргумент, °.

Если принять

$$a = 2A/d_1; \tag{86}$$

$$z \approx n_2/n_1, \tag{87}$$

то можно записать

$$V_{рез} = \frac{\pi n_2}{1000}\left(d_1/z + d_2\,\frac{a\cos\varphi - 1}{a-1}\right). \tag{88}$$

Относительное максимальное изменение скорости резания

$$\Delta V = \frac{V_{max} - V_{min}}{V_{max}}\,100\%. \tag{89}$$

Заметим, что в формуле (88) максимальное значение скорости резания имеет место при $\varphi = 0$, а следовательно, $\cos\varphi = 1$. Наименьшее значение скорости резания будет в случае, когда φ достигает максимального значения, определяемого кинематическими параметрами процесса. Аналитические выражения для V_{max} и V_{min} [29]

$$V_{max} = \frac{\pi n_2}{1000}\left(\frac{d_1}{z} - d_2\right); \tag{90}$$

$$V_{min} = \frac{\pi n_2}{1000}\left\{\frac{d_1}{z} - d_2\left[1 - \frac{(a-1)(k^2-1)}{a}\right]\right\}. \tag{91}$$

Более подробно преобразование формул (90) и (91) излишне, тем более что, как правило, кроме определения относительного изменения скорости резания необходимо знать максимальное абсолютное ее значение.

Программа 48/34. Вычисление скорости резания и ее относительного изменения при фрезеровании резьбы попутным методом:

П1	1	–	ИП2	×	ИП3	XY	÷	П4	1
ПП	36	ИП1	1	–	ИП2	x^2	1	–	×
ИП1	÷	1	XY	–	ПП	36	–	XY	÷
С/П	ИП1	ИП7	+	БП	00	ИП3	×	ИП4	ИП5
÷	XY	–	π	×	ИП6	×	1	ВП	3
÷	В/О								

Содержание регистров памяти:

П1 – относительное межцентровое расстояние a;

П2 – коэффициент глубины резания k;

П3 – диаметр заготовки d_2, мм;

П4 – диаметр инструмента d_1 (получается в результате счета по программе), мм;

П5 – число зубьев инструмента z;

П6 – n_2 – частота вращения заготовки, об/мин;

П7 – приращение относительного межцентрового расстояния Δa.

Остальные регистры памяти в данной программе не использованы.

Перед началом счета в регистр X (на индикацию) вводят требуемое начальное значение относительного межцентрового расстояния и нажимают клавиши В/О и С/П. Последующие расчеты требуют однократного нажатия клавиши С/П после индикации относительного изменения скорости резания на входе и выходе инструмента.

Попутно отметим, что приведенная программа может служить для расчета абсолютного значения скорости резания. Для этого по адресу 27 необходимо записать команду С/П. В регистре X при этом будет абсолютное значение минимальной скорости резания V_{min}, в регистрах Y, Z, T стека находится значение максимальной скорости резания. Его можно вывести на индикацию (в регистр X) нажатием клавиши XY. Затем нажатием клавиш XY, "–" и С/П продолжить вычисления. С помощью этой же программы можно моделировать процесс и строить графики зависимостей $V_{рез} = F(a)$; $V_{рез} = F(k)$; $\Delta V = F(a)$ и одновременно находить диаметр инструмента d_1 при любом относительном межцентровом расстоянии и заданном диаметре заготовки. Диаметр инструмента

Таблица 15

Результаты математического моделирования процесса вихревого резьбофрезерования

a	V_{min}	V_{max}	ΔV	d_1
1,05	76,07142	75,398222	–8,9285659–03	64,
1,10	26,417938	25,13274	–5,1136406–02	32,
1,15	10,221558	8,3775794	–2,201088 –01	21,333333
1,20	0,	2,3561944	1,	16,
1,25	– 2,1991143	– 5,0265482	5,625 –01	12,8
1,30	– 5,115158	– 8,3775812	3,8942309–01	10,666666
1,35	– 7,1059846	–10,771175	3,4027763–01	9,1428571
1,40	– 8,5272794	–12,56637	3,2142862–01	8,
1,45	– 9,5752368	–13,962636	3,1422416–01	7,111111
1,50	–10,367255	–15,079644	3,1250001–01	6,4
1,55	–10,977148	–15,993563	3,1365212–01	5,8181818
1,60	–11,453723	–16,755161	3,1640627–01	5,3333333
1,65	–11,830403	–17,39959	3,2007576–01	4,9230769
1,70	–12,130773	–17,951958	3,2426462–01	4,5714285
1,75	–12,371892	–18,430677	3,2873371–01	4,2666666
1,80	–12,56637	–18,849556	3,3333336–01	4,
1,85	–12,723701	–19,219154	3,3796768–01	3,7647058
1,90	–12,851134	–19,547687	3,4257521–01	3,5555555
1,95	–12,954299	–19,848637	3,4711541–01	3,368421
2,00	–13,037609	–20,106192	3,5156249–01	3,2

автоматически получается в регистре П4. Для построения зависимостей необходимо по адресам 00 и 31 изменить номер адресуемого регистра памяти. Им должен быть регистр, в котором записана исследуемая величина.

Результаты расчетов для заготовки $d_2 = 4$ мм; $z = 4$; $n_2 = 2000$ об/мин; $k = 1{,}25$; $a = 1{,}05$; $\Delta a = 0{,}05$ приведены в табл. 15.

Интересно отметить, что для $z = 4$, после $a = 1{,}2$, знак скорости резания изменился на противоположный (знак минус) при одном и том же направлении вращения заготовки. Это свидетельствует о том, что скорость резания направлена в сторону, противоположную нашему предположению (по направлению вращения инструмента), и образуется за счет превышения окружной скорости заготовки в месте контакта над окружной скоростью вершины инструмента. Здесь необходимо изменить положение передней поверхности инструмента и вращать его в сторону задней поверхности. Другими словами, математическая модель выявила процесс попутного точения. Необходимо сказать, что уменьшение скорости на входе-выходе характерно только для попутного точения, что также выявилось при моделировании с помощью микрокалькулятора.

В этом примере проявляется свойство математической модели быть мощным инструментом анализа поскольку абсурдный на первый взгляд результат — отрицательная скорость или отрицательное уменьшение скорости — имеет вполне законное объяснение, его необходимо только найти.

ГЛАВА V.

ОБРАБОТКА РЕЗУЛЬТАТОВ ЭКСПЕРИМЕНТАЛЬНЫХ ИССЛЕДОВАНИЙ

1. СТАТИСТИЧЕСКАЯ ОБРАБОТКА РЕЗУЛЬТАТОВ ЭКСПЕРИМЕНТА

В процессе внедрения технологических процессов или при исследованиях действующих технологических процессов возникает потребность по результатам сравнительно небольшой выборки каких-либо физических величин сделать предположение о поведении всей совокупности этих физических параметров. Наибольшее влияние на правильность выводов может оказать знание математической статистики. В других случаях необходимо какой-либо совокупности величин поставить в соответствие другую совокупность и выяснить, имеется ли между ними какая-либо связь или нет. Так, например, определить влияют ли на рассеяние размеров деталей до и после какой-либо технологической операции параметры самой операции, и если влияют, то как?

На этот и подобные вопросы может ответить математическая статистика. Причем, поскольку в статистике имеют дело, как правило, со значительными массивами данных, то обработка этих массивов вручную всегда затруднительна и трудоемка. Если необходимо все-таки провести статистическую обработку результатов исследования, то такая работа на микрокалькуляторе даже с небольшим массивом данных требует очень больших затрат времени и сосредоточенности.

При подсчете статистических характеристик массива данных всегда приходится все числа подвергать обработке по одинаковым алгоритмам, причем каждое число необходимо ввести в микрокалькулятор. Опыт работы с микрокалькуляторами показывает, что при вычислениях такого рода возникают две разновидности ошибок, вызванные усталостью оператора. Первая ошибка — это ошибка в номере числа или пропускание каких-либо чисел, а вторая — ошибка в порядке числа, например, вместо 10,1 оказываются введенными 101 или 1,01. Такие ошибки можно полностью устранить подсчетом порядкового номера числа, исключив ошибку в пропуске или повторном наборе. Во избежание второго вида ошибок нужно в программу ввести анализ значения вводимого числа.

Анализ значения числа удобен в особенности при исследовании точности технологических процессов или других явлений, где исследуемая величина имеет размах, не превышающий заранее обусловленные пределы. При отсутствии заданных пределов исследуемой величины необходимо предварительно просмотреть список исходных данных и определить максимальное изменение. Для анализа значения числа в микрокалькуляторе можно использовать формулы

$$\frac{x_{max} - x_{min}}{x_{max} + x_{min}} = \delta; \tag{92}$$

$$\frac{x_{max} + x_{min}}{2} = x_{ср}; \tag{93}$$

$$\left|\frac{x_i}{x_{ср}} - 1\right| - \delta = \left|\frac{x_i - x_{ср}}{x_{ср}}\right| - \delta \leqslant 0, \tag{94}$$

где x_{max} — максимальное значение исследуемого параметра; x_{min} — минимальное значение исследуемого параметра; δ — наибольшая относительная погрешность; $x_{ср}$ — среднее арифметическое между максимальным и минимальным значением; x_i — текущее значение исследуемого параметра.

Применение проверки вводимого числа по формуле (94) позволяет исключить заведомо большие ошибки. Так, например, если $\delta = 0{,}1$ (10%), при максимально возможной ошибке на вводимом числе погрешность математического ожидания для 100 чисел будет в 100 раз меньше, т. е. 0,001 (0,1%). Если ошибочно (в пределах максимальной относительной погрешности) введены несколько чисел, то ошибка может даже уменьшиться вследствие разного знака погрешностей.

Подсчет порядкового номера вводимого числа позволяет также избавиться от грубых ошибок, так как одно лишнее или невведенное число при массиве из 100 чисел дает погрешность результата $\pm A\%$, где A — среднее значение вводимого числа, в то время как при максимальной ошибке δ на одном числе при вводе всех чисел погрешность составит $0{,}01\delta = \delta\%$. Это позволяет получить достаточно точный результат, не зависящий от усталости и внимания оператора.

2. ОПРЕДЕЛЕНИЕ ВЫБОРОЧНОЙ СРЕДНЕЙ И МОМЕНТОВ ВЫСШИХ ПОРЯДКОВ

Оценкой математического ожидания (генеральной средней) является выборочная средняя $\bar{x}$, определяемая по формуле (ГОСТ 11.004–74)

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i. \tag{95}$$

Как видим, формула несложная, однако анализ отклонения числа и подсчет номеров усложняют вычислительный алгоритм. Поэтому программу начнем с оценки максимального отклонения числа по формуле (94). Если число выдержало эту проверку, просуммируем его и увеличим содержимое счетчика на единицу, если случайно набрано число, выходящее за пределы максимальной относительной погрешности δ, индицируем содержание счетчика со знаком минус и суммировать число не будем. После ввода всех чисел с помощью оператора цикла перейдем к определению выборочной средней.

Программа 49/34. Вычисление выборочной средней:

С/П	↑	ИП1	–	вх	÷	x^2	ИП5	XY	–
x<0	16	ИП4	/–/	БП	00	XY	ИП2	+	П2
КИП4	ИП4	L0	00	ИП2	ИП4	÷	П6	С/П	

Содержание регистров памяти:

П1 – среднее арифметическое значение $x_{ср}$ по формуле (93);

П2 – регистр-сумматор x_i; перед вычислением записать нуль;

П4 – регистр-счетчик набранных чисел, использовано свойство регистров П4, П5, П6 при косвенной адресации увеличивать содержимое на единицу; перед вычислением записать нуль;

П5 – квадрат наибольшей относительной погрешности δ^2 по формуле (92);

П6 – выборочная средняя $\bar{x}$;

П0 – счетчик цикла; перед вычислением вписать количество чисел в массиве.

Остальные регистры памяти не используются. По адресу 08 записан оператор обмена регистров стека с тем, чтобы использовать по адресу 10 оператор $x < 0$. Замена оператора 10 на $x \geqslant 0$ приводит к тому, что при $x_i = x_{max}$ или $x_i = x_{min}$ микрокалькулятор его не принимает.

В процессе работы по приведенной программе микрокалькулятор выдает пользователю служебные сообщения, закодированные числами, поэтому в таком случае можно рассматривать режим работы как диалоговый. При индикации целого положительного числа – номера последнего правильно введенного числа – оператору необходимо ввести следующее число. При индикации целого отрицательного числа следует повторно ввести число с номером, на единицу большим абсолютного значения индицированного. Наконец, при индикации вещественного числа с десятичной частью – конец цикла вычислений, на индикаторе – выборочная средняя $\bar{x}$.

При проведении экспериментальных исследований обычно генеральная средняя распределения неизвестна и пользуются выборочной средней $\bar{x}$. В этом случае для несмещенных оценок моментов второго–четвертого порядка используют формулу [33]

$$\mu_k = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^k, \tag{96}$$

где k – порядок момента.

Момент второго порядка называют выборочной дисперсией и обозначают s^2. Моменты высших порядков используют для вычисления характеристики эксцесса (степени крутизны)

$$\tau = (\mu_4/\mu_2^2) - 3 \tag{97}$$

и коэффициента асимметрии

$$\alpha = \mu_3/\sqrt{\mu_2^3}. \tag{98}$$

Выборочное среднее квадратическое отклонение

$$s = +\sqrt{s^2} \tag{99}$$

является оценкой генерального среднего квадратического отклонения.

Поскольку при вычислениях по формулам (96)–(99) снова необходимо определять значение числа по такой же схеме, как и в предыдущей программе, есть возможность вынести эту проверку в подпрограмму и высчитать все статические характеристики в одной программе.

Программа 50/34. Вычисление основных статистических характеристик для массива экспериментальных данных:

Сх	П4	ПВ	ПП	77	ИПВ	+	ПВ	ИП4	L0
03	ИПВ	XY	П0	÷	П6	Сх	П4	ПВ	ПС
ПD	ПП	77	ИП6	–	↑	x^2	ИПВ	+	ПВ
XY	↑	↑	x^2	×	ИПС	+	ПС	XY	x^2
x^2	ИПD	+	ПD	ИП4	L0	21	1	–	П4
ИПВ	XY	÷	ПВ	ИПD	ИП4	÷	ПD	ИПС	ИП4
÷	ПС	ИПВ	$\sqrt{x}$	П3	вх	×	÷	ИПD	ИПВ
x^2	÷	3	–	С/П	БП	00	С/П	↑	ИП1
–	вх	÷	x^2	ИП5	XY	–	x<0	93	ИП4
/–/	БП	77	XY	КИП4	XY	В/О			

Содержание регистров памяти:

П1 – среднее арифметическое значение $x_{ср}$, определенное по формуле (93);

П3 – выборочное среднее квадратическое отклонение s;

П4 – счетчик введенных чисел;

П5 – наибольшее относительное отклонение исследуемой величины δ;

П6 – выборочная средняя $\bar{x}$;

ПВ – момент второго порядка μ_2, выборочная дисперсия s^2;

ПС – момент третьего порядка μ_3;

ПD – момент четвертого порядка μ_4.

Перед началом работы в регистр П0 следует ввести количество чисел массива и нажать клавиши В/О и С/П. После высвечивания 0 необходимо последовательно ввести все числа массива, причем индикатор укажет номер введенного числа или на ошибку в наборе. После набора всех чисел массива на индикаторе снова высвечивается 0, это означает, что следует повторить набор всех чисел еще раз для вычисления моментов высших порядков и статистических характеристик. После повторного ввода массива на индикаторе высвечивается характеристика эксцесса, в регистре Y стека хранится коэффициент асимметрии. В регистрах памяти микрокалькулятора размещаются характеристики распределения.

В некоторых случаях, когда не нужно определять моменты третьего и четвертого порядков, для вычисления дисперсии можно применить иную формулу (ГОСТ 11.004–74):

$$\mu_2 = s^2 = \frac{1}{n-1}\left[\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2\right]. \qquad (100)$$

Тогда можно обойтись однократным введением массива с вычислением выборочной средней и дисперсии на выходе из цикла ввода чисел и подсчета сумм.

Программа 51/34. Вычисление выборочной средней $\bar{x}$ и дисперсии s^2 :

П0	Сх	ПВ	П4	П2	С/П	↑	ИП1	–	вх
÷	ИП5	XY	–	x<0	20	ИП4	/–/	БП	05
XY	x^2	вх	ИП2	+	П2	XY	ИПВ	+	ПВ
КИП4	ИП4	L0	05	ИПВ	ИП2	x^2	ИП4	÷	–
ИП4	1	–	÷	ПВ	ИП2	ИП4	÷	П2	С/П

Содержание регистров памяти :

П1, П5 – аналогичны предыдущей программе $x_{ср}$ и δ соответственно;

П2 – выборочная средняя $\bar{x}$;

ПВ – выборочная дисперсия s^2 ;

П0 – счетчик цикла.

По окончании вычислений в регистрах П2 и ПВ записаны соответственно выборочные средняя $\bar{x}$ и дисперсия s^2 .

Иногда для облегчения расчетов данные группируют по классам. Значение x_i включают в k классов одинаковой длины h . Каждое значение x_i должно принадлежать к некоторому классу; крайние классы не должны оставаться пустыми. Определяют число значений m, входящих в некоторый класс j, а среднее арифметическое верхней и нижней границ класса принимают за середину класса. Иногда поступают наоборот : принимают значения величины за середины классов, а границы классов определяют от середины прибавлением и вычитанием половины интервала между соседними значениями.

При группировке данных выгоднее может оказаться программа с вводом только середины классов и количества чисел в классе.

Программа 52/34. Вычисление основных статистических характеристик данных, сгруппированных по классам :

Сх	ПА	П3	П4	ИП2	С/П	П2	С/П	П0	ИП2
x^2	вх	ИП3	+	П3	XY	ИПА	+	ПА	КИП4
L0	09	ИП4	ИП1	–	x=0	04	ИПА	ИП3	x^2
ИП4	÷	–	ИП4	1	–	÷	ПА	ИП3	ИП4
÷	П3	С/П	БП	00					

Содержание регистров памяти :

П1 – количество чисел n в обрабатываемом массиве;

П2 – значение середины класса x_j;

П3 – выборочная средняя $\bar{x}$;

ПА – s^2 – выборочная дисперсия;

П4 – счетчик введенных чисел;

П0 – счетчик чисел m в классе.

Приведенная программа работает так же, подсказывая оператору порядок ввода чисел. Однако в связи с резким сокращением количества вводимых данных принято нецелесообразным проводить анализ значений числа. После ввода программы и нажатия клавиш В/О и С/П на индикаторе высвечивается значение середины предыдущего класса. Необходимо ввести середину очередного класса и нажать клавишу С/П. Индикация этого значения подтверждает правильность ввода и готовность микрокалькулятора к приему количества чисел m, попавших в данный класс. Необходимо ввести это число и нажать клавишу С/П. После ввода всех чисел микрокалькулятор останавливается. В регистрах П3, X и на индикаторе будет значение выборочной средней, в регистре ПА и в регистре Y – значение дисперсии.

Эта программа существенно быстродейственней предыдущих, однако требует предварительной группировки массива данных по классам и не позволяет вычислить моменты третьего и четвертого порядков.

По программе 50/34 массив из 100 чисел можно обработать за 0,7–1 ч и получить все статистические характеристики. Программа 52/34 на обработку такого же массива требует 20–25 мин: 10 мин на сортировку данных по классам и 10 мин на вычисления.

Здесь важно сделать еще одно замечание. С целью уменьшения общего числа нажимаемых клавиш при вводе необходимо обрабатываемый массив чисел привести к целым числам, изменив начало отсчета и умножив их на 10^x.

Например, массив, включающий отклонения детали от цилиндричности, описывается числами 0,003; 0,007; 0,011 и т. д. Удобно все числа этого массива умножить на 10^3 и вводить их в микрокалькулятор как 3, 7, 11 и т. д. Отклонения размера составили –0,03; –0,04; +0,02 и т. д. Для такого случая оказывается удобным за начало отсчета принять –0,04 и умножить весь массив на 100. Тогда потребуется ввести : 1; 0; 6 и т. д. Важно при этом помнить о том, что при получении окончательных результатов необходимо разделить на принятый множитель выборочную среднюю и среднее квадратическое отклонение, момент второго порядка разделить на квадрат множителя, момент третьего порядка – на третью степень множителя, момент четвертого порядка – на четвертую степень множителя, коэффициент асимметрии умножить на знак множителя. После выполнения указанных действий следует вернуться к началу отсчета массива и сместить на такое же значение выборочную среднюю.

3. ПРОВЕРКА ГИПОТЕЗЫ ЭМПИРИЧЕСКОГО РАСПРЕДЕЛЕНИЯ ПО КРИТЕРИЯМ λ (КОЛМОГОРОВА), χ^2 (ПИРСОНА) И ω^2

Статистическая проверка гипотез, т. е. предположений, относящихся к эмпирическим распределениям, играет важную роль в статистических исследованиях. Если эмпирическая кривая распределения большой выборки по своему внешнему виду приближается к какому-либо теоретическому закону распределения, то возникает вопрос, можно ли данную выборку рассматривать как выборку из генеральной совокупности, имеющей распределение именно по этому закону.

Решение этого вопроса имеет важное значение для исследователя, так как знание закона распределения случайной величины позволяет извлечь из экспериментов дополнительную информацию. По результатам серии опытов можно судить о случайных и неслучайных отклонениях, выявлять причины систематических отклонений и, следовательно, принимать меры для их устранения.

Чтобы прийти к заключению хотя бы и вероятностного характера, что распределение выборки несущественно отличается от принято теоретического закона распределения, необходимо оценить сходимость эмпирического и теоретического распределений. Для такой оценки в математической статистике существует ряд критериев (критерии согласия).

Математические методы статистики в машиностроении достаточно подробно разработаны [25, 27, 33]. Нас же интересует автоматизация обработки статистической информации на микрокалькуляторах, поэтому здесь приведены только формулы для определения критериев λ, χ^2 и ω^2. Критерий Колмогорова λ находим по формулам [27]

$$\lambda = |l_j - l'_j|_{\max} \sqrt{n}; \tag{101}$$

$$l_j = \sum_{i=1}^{j} m_i/n; \quad l'_j = \sum_{i=1}^{j} m'_i/n; \tag{102}$$

$$j = 1, 2, 3, \ldots, k,$$

где n – объём выборки; l и l' – накопленные относительные частоты, соответственно, эмпирического и теоретического распределений k – число классов группировки; m_i/n, m'_i/n – относительная частота, соответственно, эмпирического и теоретического распределений.

Рассчитанное значение критерия Колмогорова сравнивается с критическим при выбранном уровне значимости (табл. 16).

Если рассчитанное значение не превышает критического, гипотезу соответствия принимают. Критерий Колмогорова рекомендуется применять для выборки с объемом $n = 100$ и выше.

Критерий Пирсона χ^2 находят по формуле [33]

$$\chi^2 = \sum_{i=1}^{k} \frac{(m_i - m'_i)^2}{m'_i} \tag{103}$$

Рассчитанное значение критерия сравнивают с критическим при выбранных уровне значимости и количестве классов группировки (см. табл. 16). При вычислении критерия χ^2 рекомендуется объединять классы с частотами $m < 5$. Иногда применяют критерий Романовского [33]

$$A = (\chi^2 - k')/\sqrt{2k'}, \tag{104}$$

где k' – число классов с учетом объединения.

Если $A \geqslant 3$, то гипотеза о согласии эмпирического и теоретического законов отвергается, в противном случае – принимается.

Критерий ω^2 вычисляют по формуле (ГОСТ 11.006–76)

$$\omega^2 = -n - 2\sum_{i=1}^{n} \left[\frac{2i-1}{2n} \ln l'_j + \left(1 - \frac{2i-1}{2n}\right) \ln(1 - l'_j)\right]. \tag{105}$$

Результаты n наблюдений случайной величины x_i располагают в порядке возрастания. Вычисляют теоретическую накопленную частоту для каждой случайной величины и суммируют данные по формуле (105) до исчерпания всего массива данных.

Таблица 16

Критические значения параметров λ, Ω^2, χ^2 для заданных уровней значимости α

Уровень значимости α	λ	Ω^2	χ^2 для количества классов группировки k													
			2	4	5	6	7	8	9	10	11	12	13	14	15
0,05	1,52	2,50	6,00	9,49	11,07	12,59	14,07	15,51	16,92	18,31	19,68	21,06	22,36	23,69	25,00
0,10	1,36	1,94	4,61	7,78	9,24	10,65	12,02	13,62	14,68	15,99	17,28	18,55	19,81	21,06	22,31
0,20	1.07	1,42	3,22	5,89	7,29	8,56	9,80	11,03	12,24	13,44	14,63	15,81	16,99	18,15	19,31
0,50	0,83	0,79	1,39	3,36	4,35	5,35	6,35	7,34	8,34	9,34	10,34	11,34	12,34	13,34	14,34
0,75	0,68	0,51	0,58	1,92	2,68	3,46	4,26	5,07	5,90	6,74	7,58	8,44	9,30	10,17	11,04
0,90	0,61	0,36	0,21	1,06	1,61	2,20	2,83	3,49	4,17	4,87	5,58	6,30	7,04	7,79	8,55
0,95	0,57	0,29	0,10	0,74	1,15	1,64	2,17	2,73	3,33	3,94	4,58	5,23	5,89	6,57	7,26
0,99	0,44	0,21	0,02	0,30	0,55	0,87	1,24	1,65	2,09	2,56	3,05	3,57	4,11	4,66	5,23

При попадании нескольких одинаковых значений x_i в один класс j вычисления по формуле (105) проводят для каждого из чисел с одинаковой накопленной относительной частотой, но с увеличением i на единицу.

Вычисленное значение критерия ω^2 сравнивают с критическим Ω^2 при выбранном уровне значимости (см. табл. 16). Первые два критерия требуют группировки и в некоторых случаях объединения классов; их удобно вычислять в одной программе.

Вычисление критерия ω^2 по каждому числу в отдельности с изменяющимся номером числа требует иного подхода, поэтому критерий ω^2 вычислим отдельной программой.

Программа 53/34. Вычисление критериев Колмогорова λ и Пирсона χ^2:

Cx	ПD	ПА	ПС	П8	П9	П7	П8	ИП2	С/П
П2	С/П	ИПD	XY	+	ПД	вх	ИП9	+	П9
ПП	69	ИП4	×	ИПС	XY	+	ПС	вх	ИП8
+	П8	5	–	x⩾0	48	ИП8	ИП9	–	x^2
ИП8	÷	ИП7	+	П7	Cx	П8	П9	ИПD	ИПС
–	x^2	$\sqrt{x}$	↑	ИПА	–	x⩾0	60	XY	ПА
L1	08	ИПА	ИП4	$\sqrt{x}$	÷	ПА	ИП7	С/П ...	В/О

Содержание регистров памяти:

П1 – счетчик циклов – число классов;

П2 – текущее значение x_j;

П4 – общее число деталей в выборке n;

П7 – критерий Пирсона χ^2;

ПА – критерий Колмогорова λ;

П8–ПD – буферные регистры.

Приведенная программа автоматически группирует классы с содержанием теоретической абсолютной частоты m'_j менее пяти в более крупные для подсчета критерия Пирсона.

Вместо многоточия вводят программу вычисления теоретической относительной частоты события x_j. После ввода в регистры памяти требуемых величин нажатием клавиш В/О и С/П готовятся буферные регистры для счета по программе. После индикации вводят значения x_j и m_j, разделяя их нажатием клавиши С/П. При последующих индикациях микрокалькулятор выводит в регистр X предыдущее значение x_{j-1}, что облегчает работу с программой. После ввода всех чисел в регистры X и П7 выводится значение критерия χ^2, а в регистры Y и ПА критерия λ.

Для вычисления критерия ω^2 выгодно организовать цикл вычисления в пределах одного класса с изменяемым числом $2i+1$.

Программа 54/34. Вычисление критерия ω^2:

Cx	ПD	П9	1	П4	ИП2	С/П	П2	С/П	П0
ПП	59	ИПD	+	ПD	ln x	П7	1	ИПD	–
ln x	П8	ИП7	ИП4	ИПА	÷	×	1	ИП4	ИПА
÷	–	ИП8	×	+	ИП9	+	П9	КИП4	КИП4
L0	22	ИП4	1	–	ИПА	–	x=0	05	ИПА
/–/	2	÷	ИП9	2	×	–	П9	С/П	...
...	...	В/О							

Содержание регистров памяти:

П0 – счетчик циклов – количество данных в j-м классе;

ПА – число, равное удвоенному объему массива $2n$;

П2 – текущее значение переменной x_j;

П4 – счетчик $2i-1$, его используют для подсчета общего количества введенных чисел;

ПD – накопленная относительная частота для j-го класса;

П9 – значение критерия ω^2;

П8, П9 – буферные регистры.

Для работы по этой программе необходимо взамен многоточия вписать в программу функцию вычисления теоретической относительной частоты по предполагаемому закону распределения случайной величины. После нажатия клавиш В/О и С/П и индикации любого числа ввести значение середины первого класса и после нажатия клавиши С/П ввести частоту появления этого события в опыте. При последующих нажатиях клавиши С/П индицируется значение середины предыдущего класса. После ввода всего массива на микрокалькуляторе высвечивается значение критерия ω^2.

4. ТЕОРЕТИЧЕСКИЕ ЗАКОНЫ РАСПРЕДЕЛЕНИЯ СЛУЧАЙНЫХ ВЕЛИЧИН

Наибольшее распространение в технологии машиностроения нашли четыре закона распределения случайной величины [33].

Закон нормального распределения. Относительная частота появления события x_i по этому закону описывается уравнением

$$l' = \frac{1}{s\sqrt{2\pi}} e^{\frac{-(x_i-\bar{x})^2}{2s^2}} \qquad (106)$$

Абсолютная частота для всех законов определяется по формуле

$$m' = l'n. \qquad (107)$$

Программа 55/34. Вычисление относительной частоты появления события x_i при распределении случайной величины по нормальному закону:

ИП2	ИП6	–	ИП3	÷	x^2	2	÷	e^x	1/x
2	π	×	$\sqrt{x}$	÷	ИП3	÷	В/О		

Содержание регистров памяти:

П2 – текущее значение случайной величины x_i;

П3 – среднее квадратическое отклонение s;

П6 – выборочная средняя $\bar{x}$.

Закон эксцентриситета. Относительная частота появления события x_i по этому закону описывается уравнением

$$l' = \frac{x_i}{s^2} e^{-x^2/(2s^2)}. \qquad (108)$$

Программа 56/34. Вычисление относительной частоты появления события x_i при распределении случайной величины по закону эксцентриситета:

ИП2	ИП3	÷	x^2	2	÷	e^x	1/x	ИП2	×
ИП3	x^2	÷	В/О						

Содержание регистров памяти аналогично предыдущей программе.

Закон модуля разности. Относительная частота появления события x_i при распределении случайной величины по закону модуля разности описывается уравнением

$$l' = \frac{1}{s\sqrt{2\pi}}\left[e^{-\frac{(x_i-\bar{x})^2}{2s^2}} + e^{-\frac{(x_i+\bar{x})^2}{2s^2}}\right]. \qquad (109)$$

Программа 57/34. Вычисление относительной частоты появления события x_i при распределении случайной величины по закону модуля разности:

ИП2	ИП6	–	ПП	А	ИП2	ИП6	+	ПП	А
+	2	π	×	$\sqrt{x}$	÷	ИП3	÷	В/О	ИП3
÷	x^2	2	÷	e^x	1/x	П0	В/О		

Содержание регистров памяти аналогично предыдущим программам. Индексом А обозначен адрес начала подпрограммы вычисления экспоненты (оператор ИП3 после В/О).

Закон равной вероятности. Относительная частота появления события x_i в интервале $[x_1; x_2]$ описывается уравнением

$$l' = 1/(x_2 - x_1) \qquad (110)$$

и является величиной постоянной. Таким образом, при вычислении функции распределения величины, подчиняющейся закону равной вероятности, необходимо извлечь из регистра памяти константу (110).

При проверке эмпирической выборки на соответствие какому-либо теоретическому закону необходимо в программу вычисления критериев согласия записать в качестве подпрограмм одну из программ 55/34–57/34.

Построение кривой теоретического распределения случайной величины. Для графической иллюстрации близости совпадения эмпирического и теоретического распределений случайной величины оказывается удобным на гистограмму эмпирического распределения нанести кривую теоретического распределения. При этом построении выгоднее применять не относительные, а абсолютные частоты, подсчитываемые по формуле (107).

Программа 58/34. Построение координат кривой теоретического распределения:

ПП	А	ИП1	×	С/П	ИП2	С/П	П2	БП	00	... В/О

Содержание регистров памяти:

П1 – объем выборки n;

П2 – текущее значение x_i;

А – адрес начала подпрограммы вычисления относительной частоты.

В качестве подпрограмм записывают одну из программ расчета относительной частоты теоретического распределения случайной величины. Удобно пользоваться этой программой после вычисления критериев. Подпрограмма вычисления относительных частот в них уже набрана, поэтому в начале программы следует записать только программу 58/34.

5. ОПРЕДЕЛЕНИЕ ПРОЦЕНТА БРАКОВАННЫХ ДЕТАЛЕЙ ПО СТАТИСТИЧЕСКОМУ АНАЛИЗУ ВЫБОРКИ

Гипотеза соответствия теоретического и эмпирических законов позволяет распространить выбранный теоретический закон распределения случайной величины на всю генеральную совокупность величин. Это дает возможность рассчитать ожидаемый процент выхода бракованных деталей. Процент брака определяют по формуле [33]

$$q = [1 - I(x_i)]\,100, \qquad (111)$$

где q – процент брака; $I(x_i)$ – функция распределения случайной величины.

Функцию распределения случайной величины для наиболее часто применяемых законов распределения можно получить по таблицам из справочников по статистике, а можно представить как

$$I(x_i) = \int_{x_0}^{x_n} f(x)dx, \qquad (112)$$

где $f(x)$ – плотность вероятностей случайной величины, описываемая уравнениями (106), (108), (109) и им подобными.

В этом случае можно воспользоваться одной из формул приближенного вычисления определенных интегралов (73)–(75). Рекомендуется применять формулу Симпсона, как имеющую на порядок выше точность, чем формулы трапеций. В качестве примера покажем программу вычисления процента бракованных деталей при распределении размеров по нормальному закону. Для этого необходимо объединить вместе две программы 46/34 и 55/34.

Программа 59/34. Вычисление вероятного процента брака при распределении размеров детали по нормальному закону:

ПП	43	П8	ИПВ	ИПD	ПВ	–	ИП0	÷	ПD
ПП	43	1	ПП	21	4	ПП	21	2	БП
13	×	ИП8	+	П8	L0	39	ИП8	3	÷
ИПD	×	1	XY	–	2	10^x	×	С/П	ИПD
ИПВ	+	ПВ	ИПВ	ИП6	–	ИП3	÷	x^2	2
÷	e^x	1/x	2	π	×	$\sqrt{x}$	ИП3	×	÷ В/О

Содержание регистров памяти:

П0 – число шагов интегрирования – число циклов (четное);

П8 – регистр-сумматор;

ПD – наименьший размер детали, при расчете – шаг интегрирования;

ПВ – наибольший размер детали, при расчете – текущее значение аргумента;

П6 – выборочная средняя $\bar{x}$, полученная в результате эксперимента;

П3 – выборочное среднее квадратическое отклонение размера s.

Число циклов интегрирования следует брать не более восьми, так как это дает сравнительно высокую точность расчета (0,1–0,2%) при небольших затратах времени (1,5–2 мин) для расчета процента брака. Если этот расчет сравнивать с вычислением процента брака при использовании таблиц интегралов вероятности, экономия времени очевидна.

Следует помнить, что в технике теоретически вычисленный вероятный процент брака менее 0,3 принимают равным нулю, поэтому для нормального закона распределения случайной величины брак, равный нулю, лежит за пределами зоны $\pm 3s$ [33]; для закона эксцентриситета – $3{,}5s$; для закона модуля разности $(3+\rho_0)s$, где $\rho_0 = \bar{x}/s$. Если допускаемые значения измеряемого параметра лежат за пределами указанных зон, необходимо считать вероятный процент брака, в противном случае констатировать, что брак отсутствует.

6. ПРИМЕР СТАТИСТИЧЕСКОГО АНАЛИЗА ТОЧНОСТИ ОПЕРАЦИИ МЕХАНИЧЕСКОЙ ОБРАБОТКИ

В качестве примера рассмотрим такую задачу [33]. Из текущей продукции автомата, обрабатывающего ролики $\phi 20_{-0,2}$, была взята выборка $n = 100$ шт. Ролики были измерены микрометром с ценой деления 0,01 мм. Ниже приведены отклонения от номинального диаметра в миллиметрах:

–0,07	–0,03	–0,04	–0,08	–0,03	–0,08	–0,09	–0,10	–0,10	–0,10
–0,13	–0,08	–0,06	–0,04	–0,04	–0,03	–0,04	–0,07	–0,11	–0,12
–0,03	–0,07	–0,08	–0,11	–0,05	–0,05	–0,07	–0,03	–0,09	–0,10
–0,11	–0,14	–0,13	–0,08	–0,12	–0,07	–0,09	–0,10	–0,11	–0,08
–0,05	–0,12	–0,07	–0,06	–0,08	–0,11	–0,10	–0,12	–0,03	–0,10
–0,08	–0,05	–0,11	–0,07	–0,05	–0,08	–0,09	–0,09	–0,09	–0,02
–0,06	–0,12	–0,05	–0,07	–0,11	–0,05	–0,08	–0,03	–0,11	–0,09
–0,11	–0,06	–0,07	–0,06	–0,06	–0,12	–0,10	–0,08	–0,09	–0,01
–0,05	–0,07	–0,06	–0,05	–0,08	–0,09	–0,04	–0,09	–0,08	–0,09
–0,07	–0,06	–0,06	–0,12	–0,05	–0,03	–0,10	–0,09	–0,09	–0,08

Перед вычислением заметим, что удобнее работать с целыми положительными числами. Для этого каждое измеренное отклонение умножим на (-100). С этим преобразованным массивом удобно будет работать в дальнейшем, но для того, чтобы перевести выборочную среднюю в миллиметры, необходимо будет ее разделить на (-100), среднее квадратическое отклонение разделить на 100. Это неудобство пересчетов будет компенсировано тем, что при вводе чисел, клавиш будет нажато приблизительно в 3 раза меньше.

Минимальное число в 100 наблюдениях – 1, максимальное – 14. Среднее арифметическое значение $x_{ср} = 7{,}5$ (запишем в регистр памяти П1). Наибольшее относительное отклонение случайной величины, определенное по формуле (92), составляет $\delta = 0{,}86666$.

Заносим квадрат числа 0,86666 в регистр памяти П5. Вводим число 100 (количество наблюдений) в регистр П0. Устанавливаем счетчик программной памяти на нуль (клавиша В/О) и нажимаем клавишу С/П. Затем последовательно введем массив из 100 чисел, после каждого нажимая клавишу С/П. После ввода всех чисел на индикаторе высветится 0, что означает, что выборочная средняя подсчитана, и микрокалькулятор готов для ввода массива и расчета моментов высших порядков. На обработку одного числа в этой части программы требуется 6–7 с, поэтому на обработку всего массива для вычисления выборочной средней потребовалось около 12 мин. Снова последовательно введем весь массив чисел.

Служебные сообщения микрокалькулятор будет выдавать точно такие же,

как и в первой части программы, однако здесь объем вычислений выше, поэтому обработка каждого числа занимает 12–13 с. После ввода всего массива на индикаторе появляется число –7,346292–01 (коэффициент эксцесса $\tau = -0{,}735$).

В регистре Y стека находится число –1,0506412–01 (коэффициент асимметрии, умноженный на знак приведения, $\alpha = 0{,}11$).

В регистре П3 размещено число 2,8797446, которое означает выборочное расчетное среднее квадратическое отклонение $s_р$. Поскольку весь массив был умножен на –100, фактическое среднее квадратическое отклонение составит $s = 0{,}0288$. Расчетная дисперсия размещена в регистре ПВ: 8,2929292. Фактическая дисперсия составляет $s^2 = 8{,}3 \times 10^{-4}$.

Время ввода массива по второй части программы составило 20–22 мин. Значения коэффициентов свидетельствуют о том, что эмпирическая кривая несколько асимметрична и уплощена по сравнению с кривой нормального распределения, причем правая часть кривой более вытянута.

Для обработки массива из 100 чисел в общем потребовалось не более 40 мин с учетом ввода программы 50/34.

При обработке этого же массива по программе 52/34 с группировкой данных по классам те же выборочную среднюю и дисперсию получили за 10 мин, но остались неизвестными моменты третьего и четвертого порядков и коэффициенты асимметрии и эксцесса.

Полученные характеристики эмпирического распределения позволяют предположить нормальность закона распределения размера деталей (малая асимметрия $\alpha = 0{,}11$). Поэтому проверим соответствие эмпирического распределения случайной величины – отклонение размера деталей – теоретическому нормальному распределению. Для этого сгруппируем все отклонения в классы, причем фактические отклонения примем за середины классов.

Значения x_j (–100) . .	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Частота события m'	1	1	3	5	10	9	11	14	13	9	9	7	2	1

После ввода в микрокалькулятор программы расчета критериев λ и χ^2 (программа 53/34) вводим в П1 число деталей в выборке $n = 100$. Затем вводим подпрограмму вычисления относительной частоты для закона нормального распределения.

В регистр П1 вносим число 14 – количество классов группировки.

Если после выполнения предыдущих программ микрокалькулятор не выключали, то в регистре П3 будет размещено выборочное среднее квадратическое отклонение $s = 2{,}879446$, а в регистре П6 – выборочная средняя $\bar{x} = 7{,}7$. Эти величины будут использованы для вычисления относительной частоты.

Попарно вводим значения x_j и частоты m'. После вычисления критериев получаем, что $\lambda = 0{,}347$ (индикация 3,469527–01) и критерий $\chi^2 = 4{,}3072898$. Аналогично может быть вычислен критерий $\omega^2 = 2{,}06155$.

Вычисление критериев λ и χ^2 заняло 5 мин. Критерий ω^2 был подсчитан за 17 мин. Критерий ω^2 можно вычислить, выполняя побочную работу, поскольку один цикл расчета длится 1–1,5 мин и ввод данных не требует напряженно-

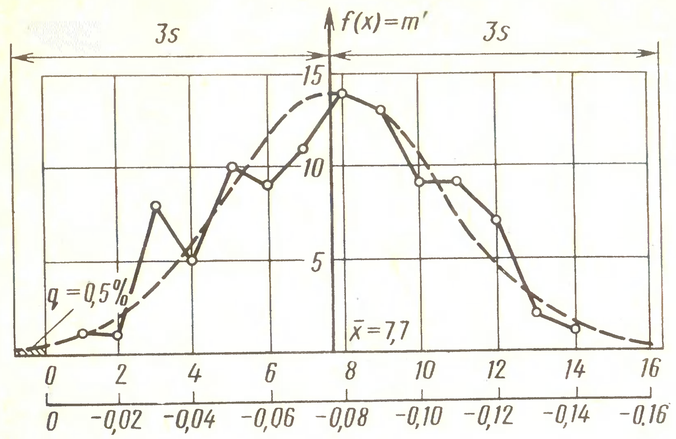


Рис. 28. Теоретическая и эмпирическая кривые распределения случайной величины

го внимания. Микрокалькулятор сам подсказывает, какое число только что обработано.

Значения критериев позволяют сделать вывод, что принятый теоретический нормальный закон распределения случайной величины можно применить к генеральной совокупности, из которой была сделана выборка объемом 100 деталей, причем критерий λ дает подтверждение гипотезы нормального распределения с вероятностью 99%, критерий χ^2 – с вероятностью 95%, а критерий ω^2 – с вероятностью 95% (см. табл. 16).

Рассчитаем вероятный процент брака исходя из условия, что предельные отклонения детали составляют: верхнее – 0, нижнее – 0,2 мм, а выборочная средняя и среднее квадратическое отклонение получены из условия предыдущей задачи.

Поскольку величина $x \pm 3s$ охватывает интервал [0,009; –0,16], меньший допуска, но смещенный так, что верхнее отклонение расположено внутри зоны $\pm 3s$, необходимо рассчитать вероятный процент брака.

Используя программу 59/34, получим, что при выполнении допуска по чертежу вероятный процент брака составит 0,52. Для интервала, в который попали все измеренные детали [–0,01; –0,14], этот процент составляет 2,4.

Очевидно, что поле $6s$ меньше поля допуска детали, поэтому необходимо изменить выборочную среднюю соответствующей поднастройкой станка, изменить настроечный размер на –0,02 мм с тем, чтобы выборочная средняя стала $\bar{x} = -0{,}097 \div -0{,}100$; при этом брак ликвидируется полностью. На рассмотренной операции имеется небольшой резерв точности $6s = 0{,}17 < \delta = 0{,}2$.

Теоретическая кривая распределения отклонений размеров деталей, построенная с использованием программы 58/34, приведена на рис. 28. Построение занимает от 3 до 5 мин.

Все приведенные расчеты, включая программирование, ввод и обработку данных при массиве из 100 чисел, потребовали 2–2,5 ч расчетов. Расчеты при этом не требуют высокой степени внимания, а следовательно, не очень утомляют.

7. КОРРЕЛЯЦИОННЫЙ АНАЛИЗ

Часто нам необходимо выяснить взаимозависимость между двумя эмпирическими величинами, которые имеют некоторые отклонения. Допускаем, что эти отклонения носят случайный характер и подчиняются закону нормального распределения. Нас также интересует, насколько точно по тем данным, которыми мы располагаем, можно провести экстраполяцию на более широкий круг данных. Например, рассеяние размеров деталей до какой-либо операции подчиняется нормальному закону, рассеяние размеров после этой операции также подчиняется нормальному закону, но имеет другие параметры. В этом случае желательно выяснить влияние данной технологической операции на изменение характеристик распределения.

Стандартной мерой близости отношения случайных величин к линейной зависимости служит коэффициент корреляции r. Этот коэффициент позволяет оценить, насколько тесно точки графика зависимости одной статистической величины от другой группируются вдоль прямой. Чем ближе точки графика к прямой, тем ближе коэффициент корреляции к ± 1. Причем $r = 1$ характеризует прямо пропорциональную зависимость, а $r = -1$ – обратно пропорциональную.

Если r близко к нулю, связать случайные величины линейной зависимостью нельзя. Коэффициент корреляции определяется по формуле [34]

$$r = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2}}, \qquad (113)$$

где x_i, y_i – случайные взаимосвязанные величины; $\bar{x}$, $\bar{y}$ – выборочные средние каждой из выборок; n – объем выборки (количество чисел массива).

Если значение коэффициента корреляции близко к ± 1, можно найти уравнение той прямой, которая наиболее вероятно связывает между собой две случайные величины.

Уравнение этой прямой (уравнение регрессии) имеет вид

$$y = bx + c; \qquad (114)$$

$$b = r\sqrt{\frac{\Sigma (y_i - \bar{y})^2}{\Sigma (x_i - \bar{x})^2}}; \qquad (115)$$

$$c = \bar{y} - b\bar{x}. \qquad (116)$$

Эти формулы легко поддаются программированию, причем выборочные средние $\bar{x}$ и $\bar{y}$ можно найти по программе 49/34 как для случайной величины x, так и для y. Для корреляционного анализа также применяем подсчет чисел, в данном случае пар чисел, и анализ значения числа. Но, поскольку здесь необходим анализ двух чисел, эту часть программы оформим подпрограммой.

Программа 60/34. Расчет коэффициента корреляции r и коэффициентов уравнения регрессий b и c:

С/П	ИП8	С/П	П8	ИП9	С/П	П9	ИП1	ИП8	ИП5
ПП	72	ИПС	ИП9	ИП6	ПП	72	ИП8	ИП2	–
x^2	вх	XY	ИПА	+	ПА	XY	ИП9	ИП3	–
x^2	вх	XY	ИПВ	+	ПВ	↻	×	ИП7	+
П7	КИП4	ИП4	L0	01	ИПА	ИПВ	×	$\sqrt{x}$	ИП7
XY	+	С/П	ИПВ	ИПА	÷	$\sqrt{x}$	×	С/П	ИП2
×	ИП3	XY	–	С/П	Сх	П7	ПА	ПВ	П4
БП	01	–	вх	÷	x^2	–	x<0	83	ИП4
/–/	БП	01	В/О						

5 Зак. 546

Содержание регистров памяти :

П1, ПС – квадрат максимального относительного отклонения величин x_i и y_i соответственно δ_x^2 и δ_y^2;

П2, П3 – выборочные средние $\bar{x}$ и $\bar{y}$ соответственно;

П5, П6 – средние значения $x_{ср}$ и $y_{ср}$ соответственно;

П8, П9 – текущие значения x_i и y_i соответственно;

П4 – счетчик количества введенных чисел;

П0 – счетчик цикла (первоначально записать объем выборки) ;

ПА, ПВ, П7 – буферные регистры;

ПD – не используется.

Работа с приведенной программой аналогична работе с программой 49/34, за исключением того, что вводится не одно число, а пара чисел x_i и y_i, разделенные нажатием клавиши С/П.

Индикация аналогична указанной программе. Однако после ввода массива данных микрокалькулятор индицирует последовательно значения коэффициента корреляции r , после нажатия клавиши С/П – углового коэффициента линейной зависимости b и после нажатия клавиши С/П – свободного члена уравнения регрессии c.

Для упрощения ввода чисел при повторных нажатиях в программу введены вызов предыдущего значения переменных. Если в новой паре переменная не изменилась, нет необходимости заново вводить ее. Достаточно нажать клавишу С/П.

В качестве примера рассмотрим отклонения размеров наружного диаметра шлицевого отверстия втулки 6×16×20 [41]. Втулка выполнена из стали 40Х. После термической обработки втулка поступает на операцию дорнования шлицевого отверстия, в результате чего размер этого отверстия изменяется. Корреляционный анализ позволяет выяснить, как влияет на отклонения размеров деталей технологическая операция дорнования. В табл. 17 приведены отклонения размеров до и после операции дорнования выборки из 50 деталей.

Поочередное введение всех пар чисел в калькулятор приводит к появлению на индикаторе коэффициента корреляции 8,6322557–01, при двух последующих нажатиях клавиш С/П получаем 5,5505443–01 и 41,432088. Среднее время обработки одной пары чисел составляет 15–20 с. Это означает, что r =0,863; b =0,555; c = 41,43. Другими словами, зависимость рассеяния размеров после дорнования от рассеяния размеров до дорнования после термической обработки достаточно близко подходит к прямо пропорциональной зависимости.

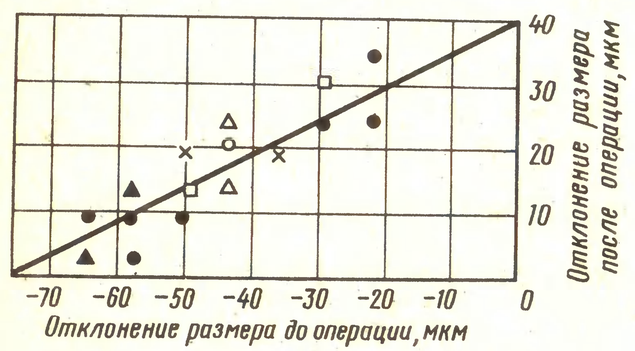


Рис. 29. Зависимости отклонений размеров деталей после операции от отклонения размеров деталей до операции:
▲ – одна деталь; ● – две детали; △ – три детали; □ – четыре детали; × – пять деталей, ○ – восемь деталей

Таблица 17

Отклонение размеров отверстия до и после операции дорнования, 10^{-2} мм

Номер детали	Отклонение размеров после термической обработки	Отклонение размеров после дорнования	Номер детали	Отклонение размеров после термической обработки	Отклонение размеров после дорнования
1	–63,5	3,5	26	–42,5	18,5
2	–63,5	8,5	27	–42,5	18,5
3	–63,5	8,5	28	–42,5	13,5
4	–56,5	3,5	29	–42,5	18,5
5	–56,5	3,5	30	–42,5	18,5
6	–56,5	8,5	31	–42,5	23,5
7	–56,5	8,5	32	–42,5	23,5
8	–56,5	13,5	33	–42,5	23,5
9	–49,5	8,5	34	–35,5	18,5
10	–49,5	8,5	35	–35,5	18,5
11	–49,5	13,5	36	–35,5	18,5
12	–49,5	13,5	37	–35,5	18,5
13	–49,5	13,5	38	–35,5	18,5
14	–49,5	13,5	39	–35,5	23,5
15	–49,5	18,5	40	–35,5	23,5
16	–49,5	18,5	41	–28,5	23,5
17	–49,5	18,5	42	–28,5	23,5
18	–49,5	18,5	43	–28,5	28,5
19	–49,5	18,5	44	–28,5	28,5
20	–42,5	13,5	45	–28,5	28,5
21	–42,5	13,5	46	–28,5	28,5
22	–42,5	13,5	47	–21,5	23,5
23	–42,5	18,5	48	–21,5	23,5
24	–42,5	18,5	49	–21,5	33,5
25	–42,5	18,5	50	–21,5	33,5

Уравнение регрессии $\bar{y} = 0{,}55\bar{x} + 41{,}43$ означает, что после операции дорнования математическое ожидание размера смещено на постоянное значение 0,04 мм и поле рассеяния размеров после операции приблизительно в 2 раза меньше, чем до операции, т. е. технологическая операция дорнования повышает точность деталей.

После выполнения программы 60/34 можно легко определить дисперсию и среднее квадратическое отклонение размеров для каждой из величин x и y. Для этого необходимо содержимое регистров ПА (для x) или ПВ (для y) разделить на содержимое счетчика чисел (П4) без единицы. Извлечение корней квадратных из полученных величин дает значение выборочного среднего квадратического отклонения $s_x^2 = 124{,}98$; $s_x = 11{,}18$; $s_y^2 = 51{,}673$; $s_y = 7{,}19$.

Влияние отклонений размеров отверстия до операции на отклонения размеров после операции показана на рис. 29.

Статистический анализ данных (см. табл. 17) потребовал 30 мин с учетом ввода программ и записи результатов.

8. АППРОКСИМАЦИЯ И ИНТЕРПОЛИРОВАНИЕ ФУНКЦИЙ

При проведении экспериментальных исследований требуется не только правильная постановка эксперимента, но и математическая обработка результатов. Так, например, если получена запись какого-либо колебательного процесса, естественно аппроксимировать осциллограмму рядом Фурье. Апериодические процессы можно аппроксимировать полиномами. Результаты аппроксимации можно использовать для улучшения математических моделей, для повышения их адекватности.

В случаях, когда необходимо определить промежуточные значения по известным экспериментальным данным с достоверными результатами, используют различные методы интерполяции.

Существует большое число методов аппроксимации и интерполяции [8, 27, 36]. Остановимся на двух из них: гармоническом анализе и интерполяции полиномами.

Автоматизация гармонического анализа

Гармоническим анализом называют замену некоторой периодической или непериодической функции $f(x)$ тригонометрической суммой:

$$s(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cdot \cos(nx) + b_n \cdot \sin(nx)]. \qquad (117)$$

Если ряд (117) сходится для всех x к функции $f(x)$, то коэффициенты ряда можно определить по формулам Эйлера–Фурье [13]:

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x)\cos(nx)\,dx; \quad n = 0;\ 1;\ 2;\ \ldots$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x)\sin(nx)\,dx; \quad n = 1;\ 2;\ 3;\ \ldots \qquad (118)$$

Тригонометрический ряд (117), коэффициенты которого определяют по формулам (118), где $f(x)$ – заданная функция, называют рядом Фурье для заданной функции $f(x)$.

Достаточным условием разложения функции в ряд Фурье на замкнутом интервале $[-\pi;\ \pi]$ является ее непрерывность на этом интервале и конечное число экстремумов.

В ряд Фурье можно разложить и непериодическую функцию. Но в этом случае ряд (117) будет сходиться к функции $f(x)$ только на интервале $]-\pi;\ \pi[$. Вне интервала и на его концах сумма ряда будет отличаться от функции.

Для вычисления коэффициентов (118) чаще всего используют замену интегралов суммами площадей трапеций [13]:

$$a_0 = \frac{1}{m} \sum_{i=0}^{m-1} y_i; \quad b_n = \frac{1}{m} \sum_{i=1}^{m-1} y_i \sin\frac{in2\pi}{m};$$

$$a_n = \frac{1}{m} \sum_{i=0}^{m-1} y_i \cos\frac{in2\pi}{m}, \qquad (119)$$

где m – четное число участков разделения периода $T = 2\pi$; $x_i = \frac{iT}{m}$ – абсциссы точек деления; $y_i = f(x_i)$ – ординаты точек деления.

Однако при вычислении интегралов (118) можно с успехом использовать формулу парабол (75). Для вычисления коэффициентов используем программу 46/34. При разложении в ряд Фурье аналитических функций в этой программе просто необходимо полученное в подпрограмме значение функции умножить на тригонометрическую функцию кратного аргумента в соответствии с формулой (118).

Если функция задана таблично, например, ординатами некоторой экспериментально полученной кривой, в подпрограмме следует ввести останов для ввода значения функции с клавиатуры. В программу необходимо добавить изменения коэффициента кратности n с тем, чтобы микрокалькулятор автоматически переходил к следующему номеру коэффициентов.

При вводе большего числа данных важно сообщать оператору, какое число необходимо ввести. И наконец, требуется блокировка от вычисления коэффициентов с номером больше допустимого разложением.

В первом приближении обычно считают, что номер коэффициента не должен быть больше половины участков разбиения периода. Количество этих участков на основе опыта можно рекомендовать выбирать не менее шестикратного числа перегибов графика функции. При этом получается достаточной высокая точность приближения и хорошая сходимость ряда (117).

Не следует прибегать к большому числу разбиений графика, поскольку на высоких частотах резко возрастает уровень "шума" и ряд (117) теряет сходимость [12].

Программа 61/34. Автоматическое вычисление коэффициентов разложения функции в ряд Фурье:

П0	П1	Сх	П4	2	π	П3	×	ИП0	÷
П2	ПП	55	П8	ИП3	/–/	П3	ПП	55	1
ПП	28	4	ПП	28	2	БП	20	×	ИП8
+	П8	L0	51	3	÷	ИП2	×	π	÷
С/П	КИП4	ИП1	П0	ИП4	2	×	–	x=0	04
÷	ИП2	ИП3	+	П3	ИП1	ИП0	–	С/П	ИП3
ИП4	×	cos x	×	В/О					

Содержание регистров памяти:

П0, П1 – число участков m разбиения периода T;

П2 – приращение аргумента – шаг разбиения;

П3 – текущее значение аргумента x_i;

П4 – номер коэффициента n;

П8 – сумматор для подсчета площадей криволинейных трапеций.

Специфика вычисления интегралов по формуле Симпсона (75) требует суммирования значения функций на концах отрезка интегрирования без умножения на число 2 или 4, в то время как промежуточные значения функции необходимо умножить на 2 или 4. Поэтому при индикации цифры 0 первоначально вводится значение ординаты на правом конце отрезка $x_m = \pi$, затем микрокалькулятор снова индицирует 0, что означает необходимость ввода ординаты вто-

рого, левого, конца отрезка $x_0 = -\pi$; последовательно возникающие цифры от 1 до $m-1$ означают номер ординаты, которую необходимо ввести.

Для работы с программой достаточно в регистр X ввести требуемое количество участков разбиения m графика функции и последовательно после индикации соответствующих цифр вводить сначала "нулевые" значения ординат графика функции, а затем остальные. Индикация положительного или отрицательного вещественного числа означает величину коэффициента, номер n которого хранится в регистре П4.

Для прекращения вычисления и сигнализации оператору о том, что коэффициенты дальше вычислять бесполезно, по адресу 50 введен оператор "÷", который вызывает индикацию некорректной операции.

Приведенная программа позволяет вычислять коэффициенты a_n разложения функции. При замене по адресу 62 оператора вычисления функции косинус на функцию синус программа будет вычислять коэффициенты b_n. Программа дает возможность рассчитать значение любого коэффициента разложения. Для этого по адресу 02 необходимо запрограммировать останов С/П и вместо Сх в процессе работы ввести в регистр X порядковый номер коэффициента, подлежащего вычислению.

Преимуществом этой программы является высокая точность вычисления коэффициентов разложения в ряд Фурье. Однако для вычисления одного коэффициента необходимо ввести последовательно все ординаты разбиения. Более простая программа получается, если в качестве формулы приближенного интегрирования принять формулу трапеций (74).

Поскольку шаг разбиения графика можно принять постоянным, эта формула примет вид

$$I(X) \approx \sum_{i=1}^{m} \frac{y_i + y_{i-1}}{2} \Delta x_i. \qquad (120)$$

Легко убедиться, что если $y_0 = y_m$, а это всегда обеспечивается при аппроксимации рядом Фурье периодических функций на интервале, равном периоду, или непериодических функций на интервале, который после аппроксимации будет рассматриваться как один период,

$$s(x_m) = s(x_0) = (y_0 + y_m)/2. \qquad (121)$$

Формулу (120) можно преобразовать:

$$I(x) = \Delta x \sum_{i=1}^{m} y_i, \qquad (122)$$

где y_i – подынтегральная функция.

Для определения коэффициентов (118) значение y_i запишется

$$y_i = f(x_i)\cos(nx) \quad \text{или} \quad y_i = f(x_i)\sin(nx)$$

в зависимости от того, какой коэффициент вычисляют. Шаг интегрирования можно определить через период T и число участков разбиения m:

$$\Delta X = T/m. \qquad (123)$$

Учитывая, что при вычислении интегралов (118) период равен $T = 2\pi$, для формулы трапеции можно записать

$$a_n = \frac{1}{\pi}\,\frac{2\pi}{m} \sum_{i=0}^{m-1} f(x_i)\cos(nx_i); \qquad (124)$$

$$b_n = \frac{2}{m} \sum_{i=1}^{m-1} f(x_i)\sin(nx_i).$$

Вычисление коэффициентов a_n и b_n можно объединить в одной программе. Кроме того, для определения значения суммы ряда (117) расчет выгоднее вести по формуле

$$s(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} A_n \sin(nx + \varphi_n), \qquad (125)$$

где $A_n = \operatorname{sgn} b_n \sqrt{a_n^2 + b_n^2}$,

$$\varphi_n = \operatorname{arctg} a_n/b_n$$

из-за того, что суммирование на микрокалькуляторе выполняется быстрее, чем вычисление тригонометрических функций. В программу целесообразно ввести вычисление значений амплитуд A_n и фаз φ_n.

Программа 62/34. Вычисление коэффициентов a_n и b_n разложения функции в ряд Фурье интегрированием по формуле трапеций и вычисление амплитуды A_n и фазового сдвига φ_n:

П0	П1	Сх	П4	Сх	ПА	ПВ	2	π	×
ИП0	+	П2	0	π	–	П3	ИП1	ИП0	–
С/П	П9	ИП4	ИП3	×	П8	cos x	×	ИПА	+
ПА	ИП9	ИП8	sin x	×	ИПВ	+	ПВ	ИП2	ИП3
+	П3	L0	17	ИПВ	ИП1	2	÷	÷	ИПА
ИП1	2	+	+	С/П	x≠0	67	КИП4	ИП1	П0
ИП4	2	×	–	x=0	04	÷	ИПВ	x^2	ИПА
x^2	+	$\sqrt{x}$	ИП1	2	÷	÷	ИПВ	x<0	81
/–/	ИПВ	+	×	ИПА	ИПВ	÷	arctg x	С/П	БП
57									

Содержание регистров памяти:

П0, П1 – количество участков разбиения периода m (число четное);

П2 – приращение аргумента Δx;

П3 – текущее значение аргумента (в радианах);

П4 – счетчик номера коэффициентов n;

П8, П9 – буферные регистры;

ПА, ПВ – коэффициенты $a_n m/2$ и $b_n m/2$ соответственно.

При работе с этой программой необходимо в регистр X ввести число разбиений периода и нажать клавиши В/О и С/П. После индикации цифры 0 следует ввести значение суммы ординат концов отрезка. При последующих индикациях нужно ввести ординату графика с номером, представленным на индикаторе. Обработка одной ординаты продолжается 10–12 с.

После ввода всех ординат в регистре X размещается значение коэффициента a_n, в регистре стека Y – значение коэффициента b_n.

Если оператору необходимо вычислить значения амплитуды и фазы, нужно

очистить регистр X (команда Cx) и нажать клавишу С/П. Это вызовет индикацию в регистре X угла сдвига фаз φ_n, а в регистре Y при этом будет находиться амплитуда A_n. При нажатии клавиши С/П, не очищая регистр X от значения коэффициента a_n или b_n, микрокалькулятор перейдет к вычислению коэффициентов гармоники с номером, на единицу большим. Если номер гармоники превышает половину числа участков разбиения, микрокалькулятор выдаст на индикацию код некорректной операции.

Подсчет суммы ряда Фурье для конкретного значения аргумента можно автоматизировать.

Программа 63/34. Суммирование синусоидального ряда Фурье с числом гармоник не более 6 :

ПD	1	3	П0	Cx	↑	XY	КИП0	ИПD	ИП0
2	÷	×	+	sin x	КИП0	×	+	ИП0	1
–	x=0	06	XY	...	...	+	СП	ИПD	π
№0	№0	÷	+	ПD	БП	01			

Содержание регистров памяти:

П1, П2 – A_1 и φ_1 соответственно;
П3, П4 – A_2 и φ_2 соответственно;
П5, П6 – A_3 и φ_3 соответственно;
П7, П8 – A_4 и φ_4 соответственно;
П9, ПА – A_5 и φ_5 соответственно;
ПВ, ПС – A_6 и φ_6 соответственно;
ПD – текущее значение аргумента.

Вместо многоточия необходимо программно ввести коэффициент A_0, вместо знаков №№ – половину требуемого числа точек разбиения периода. Желательно здесь записать то же число $m/2$, на которое разбивали участок аппроксимируемой функции при нахождении коэффициентов ряда Фурье. Тогда сумма (125) будет вычислена в узлах аппроксимации и достаточно просто посчитать сумму квадратов отклонения аппроксимируемой функции и ряда Фурье. Если число гармоник больше шести, можно автоматизировать только вычисление суммы, оставив ввод коэффициентов за оператором.

Программа 64/34. Суммирование синусоидального ряда Фурье (125) с ручным вводом коэффициентов :

Cx	П4	ПС	π	–	П3	ИП4	С/П	ИП3	ИП4
×	+	sin x	×	ИПС	+	ПС	КИП4	ИП3	ИП2
+	П3	ИП4	С/П	x=0	07	ИПС	С/П	БП	00

Содержание регистров памяти:

П4 – счетчик порядкового номера гармоники;
П3 – текущее значение аргумента;
П2 – приращение аргумента;
ПС – сумма ряда $s(x)$.

После индикации номера гармоники нужно ввести A_n в регистр Y, а φ_n в регистр X. Здесь так же, как и в программе 62/34, в зависимости от желания пользователя программа может развиваться в двух направлениях. Если после индикации номера грамоники коэффициенты с данными номерами отсутствуют, но вычисления необходимо продолжить, нужно нажать клавишу С/П без очистки регистра X и после повторной индикации номера этой гармоники ввести нули в регистры X и Y. Если коэффициентов высших гармоник нет вообще, то нужно очистить регистр X, и после нажатия клавиши С/П микрокалькулятор выдаст сумму усеченного синусоидального ряда (125).

При гармоническом анализе функций можно так же, как и в программах статистической обработки, преобразовать данные таким образом, чтобы это были целые положительные числа.

Для оценки точности аппроксимации по программе 62/34 вычисляют значения постоянного члена A_0, амплитуды A_1 и фазового сдвига φ_1 первой гармоники. По программе 63/34 или 64/34 подсчитывают несколько значений функции в узлах. Сравнив вычисленные значения функции, аппроксимированные рядом (125), с заданными значениями, принимают решение о необходимости вычисления коэффициентов следующей гармоники. Наиболее удобна работа с двумя микрокалькуляторами, один из которых вычисляет значения функции, а другой – значения коэффициентов ряда Фурье. В этом случае есть уверенность, что правильно вычислены значения всех гармоник и ни одна из существенных гармоник не упущена. Критерием погрешности вычислений является сумма квадратов отклонений заданной функции и ряда (125) в узлах аппроксимации.

При вычислении на программируемых микрокалькуляторах этот критерий достаточно трудоемок для вычисления и не может быть рекомендован к использованию после вычисления каждой гармоники.

Априорной оценкой сходимости ряда (125) к заданной функции является быстрое уменьшение амплитуды A_n с увеличением номера гармоники. Если амплитуда A_n не уменьшается, следует попробовать увеличить число участков разбиения аппроксимируемой функции. Гармоники с амплитудой, не превышающей 1–3% максимальной гармоники, в расчет не принимают. Гармоники с порядком выше третьего желательно уточнить по программе 61/34.

Пример аппроксимации рядом Фурье численного решения дифференциального уравнения (54) с нелинейностью (55). В процессе решения дифференциального уравнения (55) было замечено, что после значения времени $t = 0{,}41$ с колебательный процесс установился, поэтому можно попытаться аппроксимировать это решение тригонометрическим рядом Фурье, которое может быть одним из аналитических решений дифференциального уравнения (54) с нелинейностью (55).

Из табл. 12 видно, что период искомой функции оценивается в $T = 0{,}125$ с. Однако 12 табличных значений для аппроксимации недостаточно, поэтому, используя программы 35/34 и 37/34, решим дифференциальное уравнение (54) с нелинейностью (55) при начальных условиях, определенных для $t = 0{,}41$ с шагом $h = 0{,}005$ с. Уточненное решение показывает, что период решения составляет 0,13 с; при этом получили 26 значений периодического решения.

Для упрощения ввода чисел и обработки результатов каждое значение увеличим в 10^4 раз и округлим до целого значения. На точность вычислений это практически не повлияет, поскольку относительная погрешность в нашем случае не превысит 0,1–0,5%. Значение ординат приведены в табл. 18. Поскольку решение дифференциального уравнения является функцией периодической, время в таблице заменено номером точки. Это тоже упрощает работу с массивом чисел.

Ординаты графика численного решения уравнения (54) и суммы тригонометрического ряда в точках аппроксимации с учетом трех и четырех гармоник *Таблица 18*

Номер точки графика	На графике (см. рис. 22)	Полученные аппроксимацией			
		по методу парабол с учетом гармоник		по методу трапеций с учетом гармоник	
		трех	четырех	трех	четырех
0	33	35	17	48	33
1	42	81	60	89	62
2	158	176	171	178	162
3	325	301	316	279	307
4	467	432	454	425	451
5	557	544	555	538	558
6	611	621	611	618	612
7	634	655	633	657	634
8	629	645	630	652	629
9	598	599	604	609	605
10	543	528	548	537	557
11	469	445	464	452	478
12	382	366	366	369	378
13	291	305	286	303	288
14	236	272	251	267	240
15	253	273	268	269	251
16	324	306	321	299	309
17	391	358	380	352	371
18	427	411	421	407	427
19	435	445	434	442	439
20	417	442	420	442	418
21	374	395	381	398	374
22	309	311	317	318	314
23	229	207	229	218	238
24	137	111	129	125	151
25	54	47	47	62	72
26	40	35	17	48	33

Значение коэффициентов ряда Фурье для решения уравнения (54), заданного таблично (графически) *Таблица 19*

Номер гармоники	Коэффициенты гармоник при расчете по формулам							
	трапеций (программа 62/34)				парабол (программа 61/34)			
	a_n	b_n	A_n	φ_n	a_n	b_n	A_n	φ_n
0	720,7	–	720,7	–	716,4	–	716,4	–
1	116,0	–107,8	–158,3	–0,82	119,3	–107,8	–160,7	–0,84
2	–184,9	– 40,0	–189,2	1,35	–188,4	– 32,0	–190,8	1,40
3	11,6	– 9,6	– 15,0	–0,88	15,0	– 9,7	– 17,9	–1,00
4	– 15,0	– 22,2	– 26,8	0,59	– 18,6	– 15,6	– 28,8	0,70
5	– 1,8	– 0,4	– 1,8	1,35	– 2,2	– 0,4	– 2,2	1,39
6	6,2	– 6,5	– 9,0	0,76	2,8	– 6,3	– 6,9	–0,41
7	– 4,7	– 0,6	– 4,8	1,44	– 0,8	– 1,6	1,8	–0,46
8	6,3	– 0,1	– 6,3	–1,55	3,9	– 0,0	– 3,9	–1,56
9	– 4,9	– 0,5	– 5,0	1,47	8,1	6,9	10,6	0,87

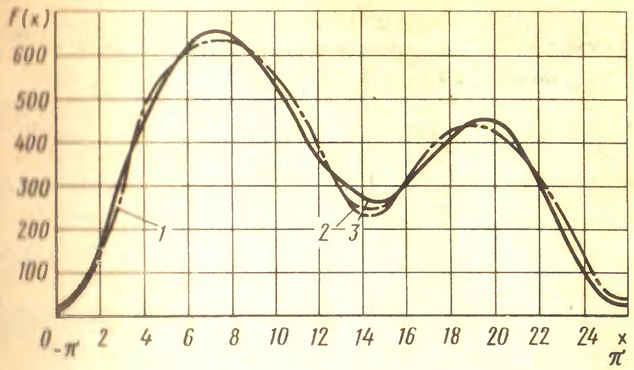


Рис. 30. Периодическое решение уравнения (54) с нелинейностью (55) и его аппроксимация рядом Фурье:
1 – периодическое решение; *2* – аппроксимация решения рядом Фурье с четырьмя гармониками; *3* – аппроксимация решения рядом Фурье с тремя гармониками

Первоначально по программе 62/34 были вычислены ориентировочные значения коэффициентов разложения в ряд Фурье, амплитуд и фаз функции. Затем для наглядности были вычислены коэффициенты для этой же функции, но с использованием программы 61/34 (табл. 19).

Оказалось, что гармоники нулевого и первого порядков отличаются несущественно, в то время как гармоники более высоких порядков имеют существенные расхождения. В табл. 18 приведены значения сумм рядов в точках аппроксимаций.

Значения, приведенные в таблице не позволяют отдать предпочтение методу трапеций или парабол при одинаковом числе членов разложения. Однако подсчет сумм квадратов отклонений показал, что для формулы трапеций при аппроксимации тремя и четырьмя членами разложения сумма квадратов отклонения равнялась соответственно $\Sigma_{T3} = 13836$; $\Sigma_{T4} = 2637$. Для формулы парабол эти суммы соответственно равнялись $\Sigma_{n3} = 11702$ и $\Sigma_{n4} = 2222$. Разница в сумме квадратов отклонений при одинаковом числе членов разложения составляет 15–20%. Таким образом, вычисление коэффициентов ряда Фурье с использованием формулы Симпсона дает точность вычисления коэффициентов на 20% выше, чем формула трапеций.

В заключение заметим, что, поскольку ординаты графика функции были увеличены в 10^4 раз, во столько же раз необходимо уменьшить амплитуды гармоник, так как они измеряются в тех же единицах, что и ординаты графика. При гармоническом анализе это пропорциональное увеличение несущественно, поэтому к первоначальным единицам измерения амплитуды следует приводить только в случае дальнейшего использования полученного ряда Фурье в аналитических приложениях. График численного решения уравнения и аппроксимирующие ряды, рассчитанные по формулам трапеций и парабол, приведены на рис. 30.

Интерполирование полиномами

Интерполирование полиномами наиболее выгодно применять при обработке результатов экспериментальных исследований, не имеющих периодических изменений. В этом случае интерполяция не позволяет экстраполировать значения функции за пределы анализируемого интервала, однако дает возможность получить значения функции в любой точке интервала. Интерполяционная формула n-го порядка аппроксимирует функцию $f(x)$ многочленом n-й степени

$Y(x)$, удовлетворяющим условиям $Y(x_k) = f(x_k)$ в $(n+1)$ точках x_k $(k = 0, 1, ..., n)$ – узлах интерполяции.

Проще всего при работе на микрокалькуляторах использовать интерполяционную формулу Ньютона [13]:

$$Y(x) = y_0 + (x-x_0)\Delta_1(x_0; x_1) + (x-x_0)(x-$$

$$-x_1)\Delta_2(x_0; x_1; x_2) + ... + \prod_{k=0}^{n-1}(x-x_k)\Delta_n(x_0; x_1; ...; x_n), \qquad (126)$$

где $\Delta_n(x_0; x_1; ...; x_n)$ – разделенные разности, которые определяют по формулам

$$\Delta_1(x_{i-1}; x_i) = (y_i - y_{i-1})/(x_i - x_{i-1});$$

$$\Delta_2(x_{i-2}; x_{i-1}; x_i) = [\Delta_1(x_{i-1}; x_i) - \Delta_1(x_{i-2}; x_{i-1})]/(x_i - x_{i-2}); \qquad (127)$$

$$\Delta_n(x_{i-n}; x_{i-n+1}; ... x_i) = [\Delta_{n-1}(x_{i-n}; ... x_{i-1}) -$$

$$- \Delta_{n-1}(x_{i-n+1}; ... x_i)]/(x_i - x_{i-n}).$$

Для построения интерполяционного многочлена строят таблицу разделенных разностей. По этой таблице можно определить степень интерполяционного многочлена. Она равна порядку тех разделенных разностей, которые практически не изменяются. Эту таблицу используют для построения интерполяционного многочлена.

При расчетах на микрокалькуляторах интерполяционный многочлен легче вычислять по схеме Горнера:

$$Y(x) \approx y_0 + (x-x_0)[\Delta_1(x_0; x_1) + (x-x_1)[\Delta_2(x_0; x_1; x_2) + ...]. \qquad (128)$$

Построение таблиц разделенных разностей требует большого числа промежуточных записей и элементарных вычислений, причем допущенная ошибка на одном из шагов сводит на нет все последующие вычисления. Применение программируемых микрокалькуляторов позволит значительно сократить процедуру расчета таблиц разделенных разностей.

Для таблиц с постоянным шагом часто вводят конечные разности. Но это дань историческим традициям, так как разделенные разности не менее удобны в расчетах, чем конечные разности [12].

Заметим, что для вычисления разделенных разностей при расчете на микрокалькуляторах с небольшим объемом памяти удобнее пользоваться не значениями x_i, а разностями

$$\Delta x = x_i - x_{i-1}; \text{ так, } x_1 - x_0 = \Delta x_0;$$

$$x_2 - x_1 = \Delta x_1;$$

$$x_2 - x_0 = x_2 - x_1 + x_1 - x_0 = \Delta x_0 + \Delta x_1;$$

$$x_3 - x_0 = \Delta x_0 + \Delta x_1 + \Delta x_2 \text{ и т. д.}$$

При расчете разделенных разностей несколько раз выполняется одна и та же процедура, но с различными регистрами памяти; ее можно описать одним фрагментом программы с косвенными обращениями к регистру памяти и к подпрограммам.

Программа 65/34. Расчет таблицы разделенных разностей порядка до $n = 5$:

5	П1	8	П0	КИП0	КИП0	XY	–	ИП0	2
+	П0	XY	КП0	L1	04	8	П1	1	4
П0	5	П2	0	ПП	81	С/П	L2	23	7
9	П8	4	ПП	45	3	ПП	45	2	ПП
45	1	ПП	45	С/П	П2	8	П1	1	4
П0	ИП2	0	КПП8	С/П	↻	XY	5	XY	–
вх	XY	ИП1	+	П1	XY	L2	52	ИП8	2
–	П8	В/О	КИП1	+	КИП1	+	КИП1	+	КИП1
+	КИП1	+	КИП0	КИП0	XY	–	XY	÷	ИП0
2	+	П0	XY	КП0	В/О				

Содержание регистров памяти:

П0 – регистр косвенного обращения к регистрам памяти y_i, $\Delta_1(x_1, x_2)$ и т. д.;

П1 – регистр косвенного обращения к регистрам памяти $x_i(\Delta x)$;

П2 – первоначально записывают x_5, затем в процессе счета регистр является счетчиком цикла;

П3 – первоначально записывают x_4, затем в процессе счета Δx_4;

П4 – первоначально записывают x_3, затем в процессе счета Δx_3;

П5 – первоначально записывают x_2, затем в процессе счета Δx_2;

П6 – первоначально записывают x_1, затем в процессе счета Δx_1;

П7 – первоначально записывают x_0, затем в процессе счета Δx_0;

П8 – первоначально записывают y_5, затем в процессе счета хранится адрес начала подпрограммы;

П9 – первоначально записывают y_4, затем после всего счета в нем хранится разделенная разность $\Delta_1(x_5, x_4)$;

ПА – первоначально записывают y_3, затем после всего счета в нем хранится разделенная разность $\Delta_2(x_5, x_4, x_3)$;

ПВ – первоначально записывают y_2, затем после всего счета в нем хранится разделенная разность $\Delta_3(x_5, x_3, x_2)$;

ПС – первоначально записывают y_1, затем после всего счета в нем хранится разделенная разность $\Delta_4(x_5, x_4, x_3, x_2, x_1)$;

ПD – первоначально записывают y_0, затем после всего расчета в нем хранится разделенная разность $\Delta_5(x_5, x_4, x_3, x_2, x_1, x_0)$.

При этом оператор нажимает клавишу С/П и выписывает в таблицу результаты, появляющиеся на индикаторе. Микрокалькулятор автоматически использует вновь полученные значения для вычисления разностей высоких порядков, поэтому ошибки ввода промежуточных значений исключены.

В качестве примера рассмотрим интерполирование экспериментальной зависимости полиномом Ньютона.

В процессе исследования производительности процесса абразивной доводки шарикоподшипников [15] с наложением ультразвука выявили некоторую зависимость снятого припуска от времени доводки. Эту зависимость предполагается включить в математическую модель производительности процесса абразивной доводки. В этом случае ее необходимо описать математически с тем, чтобы можно было рассчитать снятый припуск при любых значениях времени доводки. С помощью программы 65/34 построим таблицу разделенных разностей (табл. 20).

Разделенные разности экспериментальной зависимости *Таблица 20*

x_i	y_i	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5
0	0	2,0	–0,0450	0,00153	–0,00014	0,0000082
10	20	1,1	–0,0067	–0,00267	0,00019	
20	31	1,0	–0,0600	0,00300		
25	36	0,4	0			
30	38	0,4				
40	42					

Таблица разделенных разностей показывает, что с достаточной для практики точностью можно применить многочлен четвертого порядка. Используя верхнюю строку таблицы разделенных разностей, запишем интерполяционный многочлен по схеме Горнера:

$$Y(x) = x\,[2{,}0 + (x-10)\,[-0{,}045 + (x-20)\,[1{,}53\cdot 10^{-3} + (x-25)\,(-1{,}4\cdot 10^{-4})]]]. \qquad (129)$$

Вычисление значения функции $Y(x)$ в узлах интерполяции по формуле (129) дает точность в пятой значащей цифре.

В заключение необходимо сделать замечание. В некоторых случаях разброс точек графика вызван погрешностями измерений или влиянием других независимых случайных величин. При этом интерполяция может не привести к успеху: конечные разности с увеличением порядка не будут выравниваться.

Для обработки данных подобных экспериментальных зависимостей необходимо применить методы математической статистики или использовать программы сглаживания экспериментальных зависимостей [39].

ГЛАВА VI.

ТЕХНИКО-ЭКОНОМИЧЕСКИЕ РАСЧЕТЫ

Расчеты мощностей цехов, участков, производств проводятся достаточно часто и требуют большого количества элементарных операций. Подобные расчеты выполняют всякий раз, когда необходимо проверить "узкое место" на предприятии, рассчитать пропускную способность участка, годовую численность работающих, оценить возможность замены продукции, выпускаемой производственным подразделением, и т. д. Эти расчеты требуют максимума внимания и минимума квалификации. Программируемый микрокалькулятор при таких расчетах использовать несомненно выгодно.

1. РАСЧЕТ МОЩНОСТИ РЕМОНТНО–МЕХАНИЧЕСКОГО ЦЕХА МАШИНОСТРОИТЕЛЬНОГО ПРЕДПРИЯТИЯ

Система планово-предупредительного ремонта (ППР), существующая на каждом предприятии нашей страны, предусматривает систематическое принудительное ремонтное обслуживание оборудования согласно графику ППР, а

не работу оборудования до полного физического изнашивания или аварийного состояния. Это позволяет избежать аварийных остановок оборудования и планировать трудоемкость ремонтных работ, численность ремонтных цехов, потребное количество запасных частей и материалов. Основой системы ППР является рекомендательный цикл ремонтного обслуживания данной группы оборудования и замена конкретной модели оборудования ее ремонтным эквивалентом, названным категорией ремонтной сложности.

Полный межремонтный цикл $T_ц$ включает определенный период времени в месяцах между капитальными ремонтами. Этот период зависит от вида оборудования, условий его работы и других причин. Внутри цикла также через определенные интервалы времени должны быть принудительно проведены средний и текущий ремонт и осмотры. Время между средними ремонтами обозначают $T_{ср}$ и измеряют в месяцах. Аналогично измеряют период между текущими ремонтами $T_т$ и межосмотровый период $T_о$.

Зная межремонтные и межосмотровые периоды T_i, можно определить количество ремонтов разного вида и осмотров за весь период между капитальными ремонтами (за цикл). Естественно, что за цикл необходимо провести один капитальный ремонт:

$$N_{к.ц} = T_ц/T_ц = 1.$$

Количество средних ремонтов определяют как отношение циклового периода к периоду между средними ремонтами за вычетом единицы (количества капитальных ремонтов):

$$N_{ср} = T_ц/T_{ср} - 1.$$

Количество текущих ремонтов определют как отношение циклового периода к периоду между текущими ремонтами за вычетом количества средних и капитальных ремонтов, так как при проведении средних и капитальных ремонтов текущий ремонт не проводят:

$$N_т = T_ц/T_т - N_{ср} - 1 = T_ц/T_т - (T_ц/T_{ср} - 1) - 1 = T_ц/T_т - T_ц/T_{ср}.$$

Аналогично

$$N_о = T_ц/T_о - N_{ср} - N_т - 1 = T_ц/T_о - (T_ц/T_{ср} - 1) - (T_ц/T_т - T_ц/T_{ср}) - 1 = T_ц/T_о - T_ц/T_т.$$

В качестве планового периода принимают обычно год, поэтому необходимо пересчитать полученные количества различных видов ремонтного обслуживания на год умножением на коэффициент приведения от цикла к году:

$$N_{iг} = N_{iц}12/T_ц.$$

При больших циклах и межремонтных периодах может получиться, что по какой-либо единице оборудования один год будет отличаться от другого по числу единиц ремонтного обслуживания. Однако при расчете графика ППР обычно рассчитывают количества ремонтов по всему предприятию в целом, и эта неравномерность, возникающая на одной единице, усредняется по всему предприятию.

Для абстрагирования от различных марок, моделей и модификаций оборудования используют условные единицы ремонтной сложности. За эталон обычно принимают токарно-винторезный станок 16К20, который является станком

средней сложности и содержит 11 условных ремонтных единиц. Для других видов оборудования категория ремонтной сложности и межремонтные периоды и циклы определяют по справочникам. Различные условия эксплуатации, степень использования, коэффициент сменности можно учесть системой коэффициентов.

Общее количество ремонтов различного вида умножают на число единиц данного вида оборудования и число ремонтных единиц этого вида оборудования; при этом получают условное количество ремонтов каждого вида, выраженное в ремонтных единицах. Это абстрагирование позволяет просуммировать различные виды ремонтного обслуживания по всем видам оборудования в целом по предприятию. По сравочникам определяют трудоемкость, материалоемкость и другие показатели на одну ремонтную единицу. Поэтому, зная общее количество ремонтных единиц для проведения капитальных, средних, текущих ремонтов и осмотров, можно получить численность ремонтно-механического цеха, рассчитать потребное количество ремонтного оборудования, необходимые материалы на ремонты и прочие показатели. Наиболее трудоемким в этих расчетах является подсчет ремонтных единиц по видам ремонтов и видам оборудования и суммирование их по участкам, цехам, предприятию в целом.

Из анализа формул для подсчета количества разного рода ремонтов видно, что вычисляются они аналогично, но с различными значениями, поэтому здесь удобно использовать косвенную адресацию.

Программа 66/34. Расчет общего количества ремонтного обслуживания в единицах ремонтной сложности :

4	П1	1	ПС	5	П0	ИП5	ИП7	×	ИП8
ИП9	ИПА	×	×	×	↑	БП	27	ИП6	КИП0
÷	ИПС	XY	ПС	XY	–	×	ИПВ	ИП6	÷
×	ИПD	÷	С/П	↻	L1	18	0	С/П	БП
00									

Содержание регистров памяти:

П0 – регистр косвенной адресации;

П1 – количество циклов вычислений – четыре – соответствует количеству видов ремонтного обслуживания;

П2 – межосмотровый период $T_о$, месяц,

П3 – период между текущими ремонтами $T_т$, месяц;

П4 – период между средними ремонтами $T_{ср}$, месяц;

П5 – количество единиц данной группы оборудования;

П6 – межремонтный цикл $T_ц$, месяц;

П7 – категория ремонтной сложности оборудования в условных единицах ремонтной сложности;

П8, П9, ПА – различные коэффициенты – сменности, условий работы, использования и т. д.;

ПВ–12 – число месяцев в году;

ПС – буферный регистр;

ПD – в регистр записывают единицу, если вычисляют годовую сумму условных ремонтных единиц; если вычисляют дневную сумму записывают число рабочих дней в году.

Таблица 21

Пример расчета суммы условных ремонтных единиц

Наименование оборудования	Количество физических единиц	Условные ремонтные единицы	Количество условных единиц ремонтов			
			капитальных	средних	текущих	осмотров
Токарно-винторезный станок 16К20	121	11	334,1	1002,4	1336,45	5346,0
Токарный автомат 1Б10П	18	8	43,7	131,2	175,0	699,0
.....	...	...	...	...	...	...
Всего по участку	...	...	...	...	...	...
Всего по цеху	...	...	...	...	...	...

Иногда для расчетов необходимо знать количество ремонтов соответствующих видов не в условных ремонтных единицах, а в физических единицах. В этом случае в регистр П7 записывают единицу.

Результаты расчетов удобно записывать в таблицу, совмещая их с записью исходных данных (табл. 21).

Если предварительно подсчитать произведение количества физических единиц на категорию ремонтной сложности и систему коэффициентов, можно высвободить несколько регистров памяти и автоматически вычислять в каждой графе табл. 21 сумму условных единиц по участку, цеху и т. д.

Программа 67/34. Расчет общего количества ремонтного обслуживания с подсчетом сумм по графам таблицы :

4	П1	1	ПС	5	П0	ИП5	БП	18	ИП5
ИП6	КИП0	+	ИПС	XY	ПС	XY	–	ИПВ	ИП6
+	×	ИПD	+	ИП0	5	+	П0	XY	КИП↑
+	КП↑	ИП0	5	–	П0	L1	09	0	С/П

Содержание регистров памяти :

П0 – регистр косвенной адресации;

П1 – счетчик циклов вычислений (ввести число 4) ;

П2, П3, П4 – $T_о$, $T_т$, $T_{ср}$ – межремонтные периоды в месяцах;

П5 – произведение количества физических единиц данного вида оборудования на его категорию ремонтной сложности и коэффициенты;

П6 – межремонтный цикл в месяцах;

П7, П8, П9, ПА – сумма условных ремонтных единиц для осмотров , текущих, средних и капитальных ремонтов соответственно. Перед вычислением эти регистры необходимо очистить.

ПВ–12 – число месяцев в году.

ПС – буферный регистр.

ПD – для расчета годовой суммы ввести единицу, для расчета дневной суммы – количество рабочих дней в году.

Приведенные программы подходят для расчета по системе ППР не только

машиностроительных предприятий, но и предприятий другой системы, например легкой промышленности, автотранспорта, поскольку система ППР построена на одинаковом подходе к любому оборудованию. Однако, например, для автотранспортных предприятий межремонтный период в месяцах заменяют межремонтным пробегом в тысячах километров. Программа от этого нисколько не изменится.

2. РАСЧЕТ ПРОИЗВОДСТВЕННОЙ ПРОГРАММЫ ЦЕХА

Для расчета производственной программы цеха или нормативной стоимости обработки необходимо просуммировать нормативную трудоемкость по каждому виду работ с учетом программы выпуска. Аналогично поступают для расчета фактической трудоемкости. Подобные элементарные операции проводят, когда необходимо рассчитать материальные, трудовые и другие виды нормативов. Здесь применяют простейший алгоритм, включающий умножение на постоянную величину, суммирование и запись суммы в определенный регистр памяти. Если промежуточные суммы не записывают в таблицах, а размещают в регистрах памяти, то наиболее часто встречаются ошибки, связанные с обращением к требуемому регистру памяти. Здесь, как и в предыдущем примере, можно использовать косвенную адресацию для автоматической записи суммы в нужный регистр, таким образом освободить оператора и уменьшить количество ошибок, возникающих при расчетах.

Программа 67/34. Автоматизированный расчет трудоемкости выпуска деталей по видам работ :

ИПС	1	+	ПС	1	0	П1	2	+	П0
С/П	ИПD	×	КИП0	+	КП↑	ИП0	L1	10	БП
00									

Содержание регистров памяти.

П0 – регистр косвенного обращения;

П1 – регистр-счетчик цикла, в него необходимо ввести число видов трудоемкости;

П2–П9, ПА, ПВ – регистры-сумматоры трудоемкостей по видам работ;

ПС – счетчики наименований введенных деталей;

ПD – число деталей одного наименования, которое планируется выпустить или уже выпущено.

При работе по этой программе в регистр ПD записывают число деталей одного наименования и нажимают клавиши В/О и С/П. Далее только клавишу С/П После индикации кода трудоемкости (цифры от 12 до 3) вводят соответствующий вид трудоемкости данной детали. Регистр ПС автоматически отсчитывает число наименований введенных деталей. Для удобства работы с этой программой нужно в таблице исходных данных поставить коды трудоемкости. Например, 12 – токарные, 11 – фрезерные, 10 – сверлильные, 9 – расточные и т. д. Если трудоемкость данного вида работ в некоторой детали отсутствует, необходимо ввести нуль или очистить регистр Х. После ввода всех деталей в регистрах П2–П9, ПА, ПВ суммируются трудоемкости по видам работ. В случае,

если видов работ не 10, а меньше, нужно по адресам 04 и 05 ввести не число 10, а то, которое необходимо.

Аналогично этой программе выполняют расчеты различных таблиц.

Приведенные программы не исчерпывают всего круга инженерно-экономических задач, они лишь показывают возможности применения программируемого микрокалькулятора в простых, но многочисленных и трудоемких расчетах с использованием больших массивов данных.

В настоящее время, когда вычислительная техника в своем развитии совершила огромный скачок, появились и появляются все новые микрокалькуляторы, персональные компьютеры с колоссальными вычислительными возможностями и возрастающими объемами оперативной памяти, немыслимо стало проводить инженерные расчеты старыми методами. Точно так же, как и полстолетия назад невозможно было представить себе работу инженера без логарифмической линейки, так и сейчас ее нельзя представить без ЭВМ.

В недалеком будущем каждое рабочее место конструктора и технолога будет оснащено вычислительным комплексом с комплектом периферийного оборудования и набором прикладных программ для решения различного рода задач, поэтому навыки программирования, математические методы, привычка "общения" с ЭВМ должны прививаться уже сегодня.

Изложенная в книге небольшая часть практических задач применения математических методов в инженерном проектировании должна послужить начальным импульсом к работе инженера с ЭВМ, тем более что 70–80% расчетно-аналитических задач технологии машиностроения могут быть успешно решены с помощью программируемого микрокалькулятора.

ПРИЛОЖЕНИЕ I

НАЗНАЧЕНИЕ КЛАВИШ МИКРОКАЛЬКУЛЯТОРОВ "ЭЛЕКТРОНИКА" Б3-34 И "ЭЛЕКТРОНИКА" МК 54

Таблица П.1

Выполняемая функция в режиме автоматического счета

Обозначение клавиши	При непосредственном нажатии клавиши	После нажатия префиксной клавиши F
1	Ввод цифры 1	Вычисление e^x
2	" 2	Вычисление $\lg x$
3	" 3	Вычисление $\ln x$
4	" 4	Вычисление $\arcsin x$
5	" 5	Вычисление $\arccos x$
6	" 6	Вычисление $\mathrm{arctg}\, x$
7	" 7	Вычисление $\sin x$
8	" 8	Вычисление $\cos x$
9	" 9	Вычисление $\mathrm{tg}\, x$
0	" 0	Вычисление 10^x
.	Ввод десятичной запятой	Кольцевые перемещения информации в стеке (↻)
/–/	Изменение знака числа или порядка	Перевод в автоматический режим работы
ВП	Ввод порядка числа	Перевод в режим программирования
Сх	Очистка регистра X	Сброс действия клавиши F
$\overrightarrow{\leftarrow}$XY	Обмен содержимым между регистрами стека X и Y	Вычисление x^y
↑	Сдвиг информации в стеке на ступеньку вверх	Вызов содержимого регистра предыдущего результата
÷	Деление содержимого регистра Y на содержимое регистра X	Нахождение обратной величины $1/x$
–	Вычитание содержимого регистра X из содержимого регистра Y	Извлечение корня квадратного $\sqrt{x}$
×	Умножение содержимого регистров X и Y	Возведение в квадрат содержимого регистра X
+	Сложение содержимого регистров X и Y	Вызов константы π в регистр X
П № или X→П № №=0, 1 ... 9, A, B C, D	Запись числа, находящегося в регистре X, в адресуемый регистр №	Организация цикла L1 (счетчик–регистр П1)
ИП № или П→X № 1=0, 1 ... 9, A, B, C, D	Считывание числа из регистра № в регистр X	Организация цикла L0 (счетчик–регистр П0)
БП	Безусловный переход	Организация цикла L2 (счетчик–регистр П2)
ПП	Пошаговое прохождение программы	Организация цикла L3 (счетчик–регистр П3)

Продолжение табл. П.1

Обозначение клавиши	При непосредственном нажатии клавиши	После нажатия префиксной клавиши F
$\vec{\text{ШГ}}$	Перемещение программы на шаг вперед	В автоматическом счете не используется
$\overleftarrow{\text{ШГ}}$	Перемещение программы на один шаг назад	То же
В/О	Очистка счетчика программы, установка его на нуль	"
С/П	Пуск и останов счета по программе	"

Таблица П.2

Выполняемая функция в режиме программирования

Обозначение клавиши	При непосредственном нажатии клавиши	После нажатия клавиши F
ПП	Переход на подпрограмму	Организация цикла (регистр–счетчик П3)
$\vec{\text{ШГ}}$	Перемещение программы на один шаг вперед	Переход по условию $x < 0$
$\overleftarrow{\text{ШГ}}$	Перемещение программы на один шаг назад	Переход по условию $x = 0$
В/О	Возврат из подпрограммы	Переход по условию $x \geqslant 0$
С/П	Останов и индикация результата	Переход по условию $x \neq 0$

Примечание. Значение остальных клавиш одинаковое, как в режиме программирования, так и в режиме автоматического счета.

Таблица П3

Обозначение клавиши	Выполняемая функция после предварительного нажатия клавиши К
$\overrightarrow{\text{ШГ}}$	Косвенный переход по условию $x < 0$ на адрес, записанный в регистре, номер которого следует за клавишей
$\overleftarrow{\text{ШГ}}$	Косвенный переход по условию $x = 0$ на адрес, записанный в регистре, номер которого следует за клавишей
В/О	Косвенный переход по условию $x \geqslant 0$ на адрес, записанный в регистр, номер которого следует за клавишей
С/П	Косвенный переход по условию $x \neq 0$ на адрес, записанный в регистре, номер которого следует за клавишей
ИП или П→Х	Косвенное обращение к регистру памяти, номер которого записан в регистре косвенного обращения
П или Х→П	Косвенная запись в регистр памяти, номер которого записан в регистре косвенного обращения
БП	Косвенный безусловный переход по модифицированному адресу. По этой команде управление передается на адрес, который записан в регистре косвенного обращения. При выполнении команды адрес регистра косвенного обращения модифицируется
ПП	Косвенный переход на подпрограмму. По этой команде микрокалькулятор переходит на выполнение подпрограммы с адреса, номер которого записан в регистре косвенного обращения. При выполнении команды адрес модифицируется. После окончания подпрограммы работа продолжается с прерванного места выполнения в вызывающей программе

Примечание. Работа с косвенными переходами возможна только при счете по программе. Косвенное обращение к памяти возможно в режиме ручной работы.

ПРИЛОЖЕНИЕ II

КОДЫ КОМАНД МИКРОКАЛЬКУЛЯТОРА В РЕЖИМЕ ПРОГРАММИРОВАНИЯ

Команда	Код	Команда	Код	Команда	Код
Ввод 0	00	ПD	4Г	КБП"7"	87
" 1	01	С/П	50	КБП"8"	88
" 2	02	БП	51	КБП"9"	89
" 3	03	В/О	52	КБП"А"	8–
" 4	04	ПП	53	КБП"В"	8L
" 5	05	Кноп	54	К БП"С"	8С
" 6	06	FX≠0	57	К БП"D"	8Г
" 7	07	FL2	58	КХ≥0"0"	90
" 8	08	FX>0	59	КХ≥0"1"	91
" 9	09	FX<0	5С	КХ≥0"2"	92
" ,	0–	FX=0	5Е	КХ≥0"3"	93
		FL0	5Г	КХ≥0"4"	94
/–/	0L	FL1	5L	КХ≥0"5"	95
ВП	0С	FL3	5–	КХ≥0"6"	96
Сх	0Г	ИП0	60	КХ≥0"7"	97
↑	0Е	ИП1	61	КХ≥0"8"	98
+	10	ИП2	62	КХ≥0"9"	99
–	11	ИП3	63	КХ≥0"А"	9–
×	12	ИП4	64	КХ≥0"В"	9L
÷	13	ИП5	65	КХ≥0"С"	9С
XY	14	ИП6	66	КХ≥0"D"	9Г
F 10^x	15	ИП7	67	КПП"0"	–0
F e^x	16	ИП8	68	КПП"1"	–1
F lg	17	ИП9	69	КПП"2"	–2
F ln	18	ИПА	6–	КПП"3"	–3
F arcsin	19	ИПВ	6L	КПП"4"	–4
F arccos	1–	ИПС	6С	КПП"5"	–5
F arctg	1L	ИПD	6Г	КПП"6"	–6
F sin	1С	КХ≠0"0"	70	КПП"7"	–7
F cos	1Г	КХ≠0"1"	71	КПП"8"	–8
F tg	1Е	КХ≠0"2"	72	КПП"9"	–9
F π	20	КХ≠0"3"	73	КПП"А"	—
F $\sqrt{x}$	21	КХ≠0"4"	74	КПП"В"	–L
F x^2	22	КХ≠0"5"	75	КПП"С"	–С
F 1/x	23	КХ≠0"6"	76	К ПП"Д"	–Г
F x^y	24	КХ≠0"7"	77		
F ↻	25	КХ≠0"8"	78	КХ=0"0"	Е0
П0	40	КХ≠0"9"	79	КХ≠0"1"	Е1
П1	41	КХ≠0"А"	7–	КХ=0"2"	Е2
П2	42			КХ=0"3"	Е3
П3	43	КХ≠0"В"	7L	КХ=0"4"	Е4
П4	44	КХ≠0"С"	7С	КХ=0"5"	Е5
П5	45	КХ≠0"Д"	7Г	КХ=0"6"	Е6
П6	46	К БП"0"	80	КХ=0"7"	Е7
П7	47	К БП"1"	81	КХ=0"8"	Е8
П8	48	К БП"2"	82	КХ=0"9"	Е9
П9	49	К БП"3"	83	КХ=0"А"	Е–
ПА	4–	К БП"4"	84	КХ=0"В"	ЕL
ПВ	4L	К БП"5"	85	КХ=0"С"	ЕС
ПС	4С	К БП"6"	86	КХ=0"D"	ЕГ

Команда	Код	Команда	Код	Команда	Код
КХ<0"0"	C0	КП "↑"	LE	КИП"↑"	ГЕ
КХ<0"1"	C1	КП "1"	L1	КИП"1"	Г1
КХ<0"2"	C2	КП "2"	L2	КИП"2"	Г2
КХ<0"3"	C3	КП "3"	L3	КИП"3"	Г3
КХ<0"4"	C4	КП "4"	L4	КИП"4"	Г4
КХ<0"5"	C5	КП "5"	L5	КИП"5"	Г5
КХ<0"6"	06	КП "6"	L6	КИП"6"	Г6
КХ<0"7"	C7	КП "7"	L7	КИП"7"	Г7
КХ<0"8"	C8	КП "8"	L8	КИП"8"	Г8
КХ<0"9"	C9	КП "9"	L9	КИП"9"	Г9
КХ<0"A"	C–	КП "A"	L–	КИП"A"	Г–
КХ<0"B"	CL	КП "B"	LL	КИП"B"	ГL
КХ<0"C"	CC	КП "C"	LC	КИП"C"	ГС
КХ<0"D"	CГ	КП "D"	LГ	КИП"D"	ГГ
КП "0"	L0	КИП"0"	Г0	F вх	0

ПРИЛОЖЕНИЕ III

НЕКОРРЕКТНЫЕ ОПЕРАЦИИ В РЕЖИМЕ ПРОГРАММИРОВАНИЯ НА МИКРОКАЛЬКУЛЯТОРЕ "ЭЛЕКТРОНИКА" Б3-34 И МЕТОДЫ ИХ ИЗБЕЖАНИЯ

Некорректные операции	Способ избежания
Смена знака результата вычисления приводит к неверному вычислению (например, операторы A, B, + /–/), причем в ручном режиме и при пошаговом прохождении программы результат верный	Изменить знак числа, входящего в операцию; заслать результат в регистр памяти и вызвать его в регистр X, изменить знак: заменить знак арифметической операции : + на – или наоборот
Вычисления вида $M/\lg x$, где M – любое число, а $1/e < x < 1$, при автоматическом счете вызывает индикацию "ЕГГОГ"	Вычислить знаменатель и затем все выражение
Выполнение операций x^y после двухместной операции, в результате которой получилось число, имеющее в восьмом разряде знак 5 или 7, приводит к неверному результату	Перед этой операцией разместить два оператора F $1/x$, F $1/x$
Вычисления тригонометрических функций больших углов : $x = 180^\circ n + a$, где $n = \pm 5, \pm 10; \pm 15$ и т. д., а $10^\circ < a < 40^\circ$ дает неверный результат во всех режимах расчетов	Вычисление тригонометрических функций больших углов следует вести при аргументе, выраженном в радианах. При необходимости в начале программы вставить программу перевода углов из градусов в радианы
Завершение подпрограммы одноместной операцией перед командой В/О ведет к неверному результату	Подпрограмму, заканчивающуюся одноместной операцией, необходимо дополнить засылкой результатов в какой-либо регистр памяти
Начало подпрограммы операцией изменения знака приводит к неверному результату	Подпрограмму начинают засылкой числа в регистр памяти и изменением знака числа
Окончание подпрограммы операцией деления, если в результате предыдущих операций возможное делимое станет равным нулю, ведет к сообщению об ошибке	Закончить подпрограмму следует операцией засылки результата в один из регистров памяти
Изменение знака константы π после ее вызова в регистр X ведет к неверному результату	Использовать буферный регистр памяти для смены знака. Использовать операцию вычитания значения π из нуля

СПИСОК ЛИТЕРАТУРЫ

1. **Андронов А.А., Витт А.А., Хайкин С.Э.** Теория колебаний. 2-е изд. М., 1981. 568 с.
2. **Бесекерский В.А., Попов Е.П.** Теория систем автоматического регулирования. М., 1966. 992 с.
3. **Васильев Ф.П.** Лекции по методам решения экстремальных задач. М., 1974. 374 с.
4. **Вирченко Н.А., Ляшко И.И., Шевцов К.И.** Графики функций: Справочник, 2-е изд. Киев, 1981. 320 с.
5. **Вульф А.М.** Резание металлов. 2-е изд. Л., 1973. 496 с.
6. **Гусак А.А.** Элементы методов вычислений. 2-е изд. Минск, 1982. 166 с.
7. **Давыденко П.М.** Организация труда технолога. Киев, 1978. 102 с.
8. **Демидович Б.П., Марон И.А., Шувалова Э.З.** Численные методы анализа. М., 1962. 376 с.
9. **Дробушевич Г.А.** Программирование на ФОРТРАНе. Минск, 1976. 272 с.
10. **Ивченко Г.И., Медведев Ю.И.** Математическая статистика: Учебное пособие для втузов. М., 1984. 248 с.
11. **Ильин В.А., Позднях Э.Г.** Основы математического анализа. М., 1971. 330 с.
12. **Калиткин Н.Н.** Численные методы. М., 1978. 512 с.
13. **Корн Г., Корн Т.** Справочник по математике для научных работников и инженеров. М., 1972. 832 с.
14. **Кройль Г.** Что умеет мой микрокалькулятор? М., 1981. 133 с.
15. **Луговой В.П., Пикус М.Ю.** Влияние ультразвуковых колебаний на процесс доводки шариков// Приборостроение. Минск, 1980. Вып. 3. С. 43–46.
16. **Микрокалькулятор** "Электроника" Б3-34: Руководство по эксплуатации. Светловодск, Кировоградской обл., 1980. 157 с.
17. **Минимизация** в инженерных расчетах на ЭВМ/ Библиотека программ/ С.Ю. Гуснин, Г.А. Омельянов, Г.В. Резников и др. М., 1981. 120 с.
18. **Основы** функционально-стоимостного анализа. Под ред. М.Г. Карпунина и Б.И. Майданчика. М., 1980. 287 с.
19. **Папшев Д.Д.** Отделочно-упрочняющая обработка поверхностным пластическим деформированием. М., 1978. 152 с.
20. **Петренко А.И.** Основы автоматизации проектирования. Киев, 1982. 295 с.
21. **Петров М.И.** Техника вычислений на микрокалькуляторе "Электроника Б3-19м". М., 1981. 167 с.
22. **Понтрягин Л.С.** Обыкновенные дифференциальные уравнения. М., 1982. 332 с.
23. **Поснов Н.Н., Поснова М.Ф.** Микрокалькуляторы "Электроника Б3" в учебной лаборатории. Минск, 1981. 192 с.
24. **Применение** ЭВМ при расчете и конструировании автомобиля/ А.И. Гришкевич, Л.А. Молибожко и др. Минск, 1978. 264 с.
25. **Пугачев В.С.** Введение в теорию вероятностей. М., 1968. 368 с.
26. **Размерный** анализ технологических процессов/ В.В. Матвеев, М.М. Тверской, Ф.И. Бойков и др. М., 1982. 264 с.
27. **Румшиский Л.З.** Математическая обработка результатов экспериментов: Справочное пособие. М., 1971. 192 с.
28. **Свидерский Э.А.** Определение величины минимального заднего угла инструмента при попутном точении// Машиностроение. 1980. Вып. 5. С. 101–106.
29. **Свидерский Э.А.** Определение скорости резания и скорости внедрения в процессах с переменной толщиной снимаемого слоя// Машиностроение. 1981. Вып. 6. С. 89–93.

30. **Свидерский Э.А.** Проектирование наладок на автоматы продольного точения при мелкосерийном характере производства// Приборостроение. 1980. Вып. 3. С. 56–60.
31. **Свидерский Э.А.** Расчет параметров затылования металлорежущего инструмента// Станки и инструмент. 1982. № 11. С. 19–20.
32. **Свидерский Э.А., Сургут Я.М.** Автоматизированное проектирование инструмента с применением программируемых ЭВМ// IV Республиканская научно-техническая конференция "Автоматический контроль и управление производственными процессами". Минск, 1983. С. 199.
33. **Солонин И.С.** Математическая статистика в технологии машиностроения. М., 1972. 216 с.
34. **Сперанс Ф.** Как обращаться с цифрами, или Статистика в действии. Пер. с англ. Минск, 1983. 271 с.
35. **Сургут Я.М., Свидерский Э.А., Котиков П.Ф.** К определению площади пятна контакта торцового резца и поверхности// IV Республиканская научно-техническая конференция "Автоматический контроль и управление производственными процессами". Минск, 1983. С. 213.
36. **Троценко В.К., Любич Ф.Д.** Инженерные расчеты на микрокалькуляторах. Киев, 1980. 380 с.
37. **Троценко В.К., Любич Ф.Д.** Радиотехнические расчеты на микрокалькуляторах: Справочное пособие. М., 1983. 256 с.
38. **Троценко В.К.** Искусство программирования программируемых микрокалькуляторов// Известия вузов. Сер. Радио-электроника. 1984. Т. 27. № 2. С. 39–46.
39. **Цветков А.Н., Гвинчванков В.А.** Прикладные программы для микроЭВМ "Электроника Б3-34", "Электроника МК-56", "Электроника МК-54". М., 1984. 175 с.
40. **Чистосердов П.С., Цумарев О.В.** Графоаналитический метод определения оптимального расположения режущего и деформирующего элементов// Машиностроение. 1980. Вып. 5. С. 41–43.
41. **Чистосердов П.С., Шадуро Р.Н.** Оптимизация выбора размеров инструмента и заготовки при дорнировании шпоночных отверстий// Технология, организация и экономика машиностроительного производства. 1981. № 9. С. 8–10.
42. **Шаумян Г.А.** Комплексная автоматизация производственных процессов. М., 1973. 640 с.
43. **Шуп Т.** Решение инженерных задач на ЭВМ. М., 1981. 183 с.
44. **Яворский П.Н., Горбацевич А.Ф.** Применение вычислительной техники в лабораторных работах по технологии машиностроения. Минск, 1978. 100 с.

ОГЛАВЛЕНИЕ

Предисловие 3

Введение 4

ГЛАВА I. Программирование на микрокалькуляторах 8

1. Характеристика и функциональные возможности микрокалькуляторов "Электроника" Б3-34 и МК54 8
2. Ввод и отображение чисел 10
3. Выполнение операций 10
4. Работа с адресуемыми регистрами памяти 13
5. Организация цепочных вычислений 14
6. Основы и особенности программирования на микрокалькуляторах 15

ГЛАВА II. Размерные расчеты при проектировании технологических процессов 29

1. Размерные расчеты деталей при групповой технологии 29
2. Организация учета деталей, обрабатываемых на токарных автоматах, и исследование возможностей обработки вновь разрабатываемых деталей на имеющейся оснастке 36
3. Размерный анализ технологических процессов 40

ГЛАВА III. Элементы автоматизации проектирования средств технологического оснащения 44

1. Методы решения трансцендентных уравнений 45
2. Приложение программ решения трансцендентных уравнений к проектированию оптимальных параметров технологических процессов и оснастки 49
3. Оптимизационные расчеты в технологии машиностроения 59
4. Приложения программ оптимизации к проектированию режимов механической обработки 74

ГЛАВА IV. Математическое моделирование 79

1. Численные методы решения дифференциальных уравнений 81
2. Построение фазовых траекторий 92
3. Численные методы расчета определенных интегралов 105
4. Приложение методов вычисления определенных интегралов при моделировании процесса поверхностного пластического деформирования 108
5. Исследование процесса вихревого фрезерования резьб 110

ГЛАВА V. Обработка результатов экспериментальных исследований 114

1. Статистическая обработка результатов эксперимента 114
2. Определение выборочной средней и моментов высших порядков 115
3. Проверка гипотезы эмпирического распределения по критериям λ (Колмогорова), χ^2 (Пирсона) и ω^2 119
4. Теоретические законы распределения случайных величин 123
5. Определение процента бракованных деталей по статистическому анализу выборки 125

6. Пример статистического анализа точности операции механической обработки 126
7. Корреляционный анализ 128
8. Аппроксимация и интерполирование функций 132

ГЛАВА VI. Технико-экономические расчеты 142

1. Расчет мощности ремонтно-механического цеха машиностроительного предприятия 142
2. Расчет производственной программы цеха 146

Приложения 148

Список литературы 154

Производственное издание

ЭДУАРД АНТОНОВИЧ СВИДЕРСКИЙ

РЕШЕНИЕ ТЕХНОЛОГИЧЕСКИХ ЗАДАЧ В МАШИНОСТРОЕНИИ С ПРИМЕНЕНИЕМ МИКРОКАЛЬКУЛЯТОРОВ

Редактор *А.В. Лысенко*
Художественный редактор *С.С. Водчиц*
Обложка художника *Е.С. Васильева*
Технический редактор *Г.Г. Семенова*
Корректор *Л.Н. Горлова*

ИБ № 5268

Сдано в набор 15.07.86. Подписано в печать 29.06.87. Т–14864.
Формат 60 X 88 1/16. Бумага офсетная № 2. Гарнитура Пресс Роман.
Печать офсетная. Усл. печ. л. 9,8. Усл. кр.-отт. 10,05. Уч.-изд. л. 11,96.
Тираж 40 000 экз. Заказ 546 Цена 60 к.

Ордена Трудового Красного Знамени издательство "Машиностроение",
107076, Москва, Стромынский пер., 4.

Отпечатано в Московской типографии № 4 Союзполиграфпрома
при Государственном комитете СССР по делам издательств, полиграфии
и книжной торговли 129041, Москва, Б. Переяславская, 46, с оригинала-макета,
изготовленного в издательстве "Машиностроение" на наборно-пишущих машинах

УВАЖАЕМЫЙ ЧИТАТЕЛЬ!

Издательство "Машиностроение" выпустит в 1987 году следующие книги по приборостроению:

ПРОИЗВОДСТВЕННАЯ ЛИТЕРАТУРА

Брусков А.М., Брусков В.М. **Конструирование зеркально-призменных оптико-механических узлов.** – 8 л.: ил. (Б-ка приборостроителя). – (В обл.): 40 к.

Волоконная оптика и приборостроение / М.М. Бутусов, С.Л. Галкин, С.П. Оробинский, Б.П. Пал; Под общ. ред. М.М. Бутусова. – 19 л.: ил. – (В пер.): 1 р. 50 к.

Высокоточные угловые измерения / Д.А. Аникст, К.М. Константинович, И.В. Мальский и др.; Под общ. ред. Ю.Г. Якушенкова. – 33 л.: ил. – (В пер.): 2 р. 10 к.

Гроссман Н.Я., Шмырев Г.Д. **Автоматизированные системы взвешивания и дозирования. Расчет, конструкция, эксплуатация.** – 24 л.: ил. – (В пер.): 1 р. 60 к.

Кинематическая точность приборных волновых передач / С.Г. Борисов, С.Н. Истомин, П.К. Попов и др. – 9 л.: ил. – (Б-ка приборостроителя). – (В обл.): 50 к.

Королев М.В., Стариков Б.П., Карпельсон А.Е. **Ультразвуковые импульсные приборы контроля прочности материалов.** – 7 л.: ил. – (Б-ка приборостроителя). – (В обл.): 35 к.

Лопухин В.А., Гурылев А.С. **Автоматизация визуального технологического контроля в электронном приборостроении.** – 19,5 л.: ил. – (В пер.): 1 р. 30 к.

Мишин В.А., Белый М.И. **Защита электроизмерительных приборов от механических воздействий.** – 11 л.: ил. – (Б-ка приборостроителя). – (В обл.): 50 к.

Яковлев П.П., Мешков Б.Б. **Проектирование интерференционных покрытий.** – 12 л.: ил. – (Б-ка приборостроителя). – (В обл.): 55 к.