

АННОТАЦИЯ

Дьячкова И.В.: «Снижение трудоемкости проектирования операций механической обработки поверхностей пространственно-сложной формы» –
Выпускная квалификационная работа –
Челябинск: ЮУрГУ, 2017. – 89 с., **21** ил.,
16 табл., **3** прил., библиографический
список – 29 наим..

В ходе выполнения выпускной квалификационной работы изучен вопрос обработки пространственно-сложных поверхностей на станках с ЧПУ; рассмотрен автоматизированный способ получения управляющих программ; разработана программа расчета погрешности размера динамической настройки при чистовом фрезеровании пространственно-сложных поверхностей со ступенчатым припуском; разработана методика инженерного проектирования, позволяющая снизить трудоемкость проектирования операций механической обработки пространственно-сложных поверхностей.

ЮУрГУ.150405.2017.147.00.00.ПЗ ВКР				
Изм.	Лист	№ докум.	Подпись	Дата
Разраб.		Дьячкова И.В.		23.05.17
Провер.		Батуев В.В.		25.05.17
Реценз.		Антонов А.А.		25.05.17
Н. Контр.				
Утверд.		Гузев В.И.		01.06.17

Снижение трудоемкости проектирования операций механической обработки поверхностей пространственно-сложной формы		
Лит.	Лист	Листов
	6	89
ЮУрГУ Кафедра ТАМ		

ОГЛАВЛЕНИЕ

АННОТАЦИЯ.....	6
ВВЕДЕНИЕ.....	9
1 ОБРАБОТКА ПРОСТРАНСТВЕННО-СЛОЖНЫХ ПОВЕРХНОСТЕЙ НА СТАНКАХ С ЧПУ (СОСТОЯНИЕ ВОПРОСА)	12
1.1 Тенденции развития современного машиностроения	12
1.2 Технология изготовления деталей пространственно-сложной формы.....	13
1.3 Проблемы в обеспечении точности	15
1.4 Постановка цели и задач	17
2 АНАЛИЗ КОНСТРУКТИВНЫХ ЭЛЕМЕНТОВ, СОСТАВЛЯЮЩИХ ПРОСТРАНСТВЕННО-СЛОЖНЫЕ ПОВЕРХНОСТИ.....	19
3 АВТОМАТИЗИРОВАННЫЙ СПОСОБ ПОЛУЧЕНИЯ УПРАВЛЯЮЩИХ ПРОГРАММ	20
3.1 Описание способа автоматизированного получения управляющих программ.....	20
3.2 Анализ рынка САМ систем	21
3.3 Анализ функциональности САМ систем	24
3.3.1 PowerMill	24
3.3.2 Mastercam	28
3.3.3 SprutCAM	29
3.3.4 ADEM	35
3.4 Выводы	37
4 РАЗРАБОТКА ПРОГРАММЫ РАСЧЕТА ПОГРЕШНОСТИ РАЗМЕРА ДИНАМИЧЕСКОЙ НАСТРОЙКИ ПРИ ЧИСТОВОМ ФРЕЗЕРОВАНИИ ПРОСТРАНСТВЕННО-СЛОЖНЫХ ПОВЕРХНОСТЕЙ СО СТУПЕНЧАТЫМ ПРИПУСКОМ	38
4.1 Постановка задачи, решаемой с помощью программы	38
4.2 Формализация поставленной задачи	38
4.3 Выбор метода решения поставленной задачи	42
4.4 Разработка алгоритма.....	45
4.5 Написание и отладка программы.....	47
4.6 Выводы по разделу	47
5 ОБРАБОТКА РЕЗУЛЬТАТОВ ВКР	49
5.1 Разработка методики инженерного проектирования.....	49

						Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

5.2	Пример применения методики инженерного проектирования	57
5.3	Оценка трудоемкости проектирования операции механической обработки по разработанной методике	59
	ЗАКЛЮЧЕНИЕ	60
	БИБЛИОГРАФИЧЕСКИЙ СПИСОК	63
	ПРИЛОЖЕНИЯ	67
	ПРИЛОЖЕНИЕ А ПРОГРАММА РАСЧЕТА ПОГРЕШНОСТИ РАЗМЕРА ДИНАМИЧЕСКОЙ НАСТРОЙКИ	67
	ПРИЛОЖЕНИЕ Б ПУБЛИКАЦИЯ ПО ТЕМЕ ВКР.....	88
	ПРИЛОЖЕНИЕ В ПРОТОКОЛ ПРОВЕРКИ НА АНТИПЛАГИАТ	89

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

ВВЕДЕНИЕ

Основным типом современного машиностроительного производства является серийное производство, примерно 80% всей продукции машиностроения обрабатывается на заводах данного типа [1]. Проектирование новых и совершенствование существующих изделий машиностроения обуславливают большой объем и частую смену номенклатуры деталей. При этом возрастает количество деталей, имеющих пространственно-сложную форму. К таким деталям относятся формообразующая оснастка (пресс-формы, штампы, модели для точного литья и т.д.), детали авиационных двигателей, лопатки турбин. Так, например, современная авиация практически на 100% состоит из машин, оснащенных газотурбинными двигателями, которые имеют в своем составе множество деталей пространственно-сложной формы (моноколеса, крыльчатки, лопатки турбин, диски компрессора и турбины, лабиринты, а также различные корпусные детали, кронштейны, фланцы и т.д.) [6, 7]. К деталям данного класса часто предъявляются высокие требования точности (от 6 качества и выше) и шероховатости ($Ra \leq 0,63$ мкм) [6, 22].

Для достижения требуемой точности технологический процесс обработки деталей пространственно-сложной формы, как правило, включает в себя три стадии [2, 3, 23]:

- черновое фрезерование;
- чистовое фрезерование;
- отделочно-доводочные операции.

Первые две стадии выполняются на станках с ЧПУ. Черновая стадия предназначена для снятия напуска и приближения конфигурации заготовки к конфигурации детали. Чистовая стадия служит для снятия ступенчатого припуска и получения точности размеров и формы готовой детали. Часто точность, предусматриваемая машинной обработкой, не может быть

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

достигнута за один чистовой переход, поэтому возникает необходимость введения дополнительных переходов.

При выполнении третьей стадии используются следующие методы доводочной обработки: Для окончательной обработки детали и достижения заданных чертежом требований по точности и шероховатости могут быть применены следующие методы доводочной обработки:

- ручная (слесарная);
- физико-химическая;
- абразивная.

Основным средством автоматизации механической обработки деталей пространственно-сложной формы являются станки с ЧПУ, а средством подготовки управляющих программ – Computer Aided Manufacture (CAM) системы.

Исходными данными для проектирования операции механической обработки в САМ системе является САД-модель детали. При проектировании операции механической обработки в САМ системе деталь разбивается на ряд конструктивных элементов, для каждого из которых технологом-программистом назначаются технологические переходы обработки. Таким образом, технолог-программист формирует траекторию движения инструмента. Выходными данными такой процедуры программирования является фактически массив положений режущего инструмента (Cutter Locations DATA, CLDATA), который затем преобразуется постпроцессором, в управляющую программу.

При разработке управляющей программы, может быть применена одна из стратегий формирования траектории движения инструмента либо их комбинация. Применяя к детали те или иные стратегии формирования траектории, можно обеспечить наиболее благоприятные условия резания и сократить долю вспомогательного времени, связанного с холостыми перемещениями, что окажет влияние на точность и производительность

									Лист
									10
Изм.	Лист	№ докум.	Подпись	Дата					

операции фрезерования. Но в то же время, решения, принимаемые на стадии проектирования операции механической обработки в САМ системе, зависят от опыта, квалификации, а порой и интуиции технолога, что оказывает существенное влияние на точность и производительность операции в целом.

Рекомендации по назначению стратегий обработки отдельных конструктивных элементов, на которые разбивается деталь в процессе проектирования операции механической обработки в САМ системе, отсутствуют. Поэтому исследования, направленные на классификацию геометрических элементов пространственно-сложных поверхностей и разработки рекомендаций по назначению стратегий их обработки являются актуальными.

						Лист
						11
Изм.	Лист	№ докум.	Подпись	Дата		

1 ОБРАБОТКА ПРОСТРАНСТВЕННО-СЛОЖНЫХ ПОВЕРХНОСТЕЙ НА СТАНКАХ С ЧПУ (СОСТОЯНИЕ ВОПРОСА)

1.1 Тенденции развития современного машиностроения

Основным типом современного машиностроительного производства является серийное производство. Примерно 80% всей продукции машиностроения обрабатывается на заводах серийного типа производства [1]. Проектирование новых и совершенствование существующих изделий машиностроения обуславливают большой объем и частую смену номенклатуры деталей. При этом возрастает количество деталей, имеющих пространственно-сложную форму. К таким деталям относится формообразующая оснастка (пресс-формы, штампы, модели для точного литья и т.д.), детали авиационных двигателей, лопатки турбин. Так, например, современная авиация практически на 100% состоит из машин, оснащенных газотурбинными двигателями, которые имеют в своем составе множество деталей пространственно-сложной формы (моноколеса, крыльчатки, лопатки турбин, диски компрессора и турбины, лабиринты, а также различные корпусные детали, кронштейны, фланцы и т.д.) [6, 7]. Классификация деталей ГТД (газотурбинных двигателей) по технологическим признакам приведена в таблице 1.1. К деталям данного класса предъявляются высокие требования точности (от 6 качества и выше) и шероховатости ($Ra \leq 0,63$ мкм) [6, 22].

Повышение требований к производительности и точности изготовления деталей пространственно-сложной формы вызывает необходимость применения новых технологий и автоматизации технологических процессов их изготовления.

В современном машиностроении для изготовления деталей пространственно-сложной формы широко применяются станки с ЧПУ, являющиеся основным средством автоматизации механической обработки. А для подготовки управляющих программ, от качества и надежности которых

						Лист
						12
Изм.	Лист	№ докум.	Подпись	Дата		

зависит точность и производительность обработки деталей на станках с ЧПУ, Computer Aided Manufacture (CAM) системы.

Таким образом, развитие современного машиностроения связано с внедрением современного автоматизированного оборудования, режущего инструмента и средств САПР, обеспечивающих высокую производительность и качество проектирования и изготовления машиностроительной продукции.

Таблица 1.1 – Классификация деталей ГТД по технологическим признакам

Параметры	Лопатки	Крыльчатки	Моноколеса	Диски компрессора и турбины	Лабиринты	Корпуса	Кронштейны	Крышки	Фланцы	Валы
1. Сложная пространственная форма (x, y, z)	■	■	■	■					■	
2. Обработка более чем по 3-м сторонам детали	■	■	■	■		■				
3. Сложная ступенчатая форма или криволинейный контур, или эферических, конических поверхностей						■		■		■
4. Лыски, выборки, ребра жесткости, торцовки, канавки, выточки				■		■		■		■
5. Стыковые и опорные плоскости, расположенные под разными углами				■		■		■		■
6. Отверстия различного типа и размера и расположения						■		■		■
7. Длинные расточки и расточки внутри детали						■		■		■
8. Точность размеров детали от 6 качества и выше	■	■	■	■		■	■	■		■
9. Детали, требующие при обработке на универсальных станках специальной оснастки и фасонных режущих инструментов	■	■	■	■		■	■	■		■
10. При обработке на станке с ЧПУ уменьшится количество применяемого режущего инструмента	■	■	■	■		■	■	■		■
11. При обработке на станке с ЧПУ уменьшится число переустановок детали	■	■	■	■		■	■	■		■

– характерный параметр;
 – переменный параметр.

1.2 Технология изготовления деталей пространственно-сложной формы

Как правило, технологический процесс обработки деталей пространственно-сложной формы включает в себя следующие стадии обработки [2, 3, 23]:

- черновое фрезерование;
- чистовое фрезерование;
- отделочно-доводочные операции.

Первые две стадии выполняются на станках с ЧПУ. Черновая стадия предназначена для снятия напуска и приближения конфигурации заготовки к конфигурации детали (рисунок 1.1). Чистовая стадия служит для снятия ступенчатого припуска и получения точности размеров и формы готовой детали (рисунок 1.2).

Часто точность, предусматриваемая машинной обработкой, не может быть достигнута за один чистовой переход, поэтому возникает необходимость введения дополнительных переходов.

Для окончательной обработки детали и достижения заданных чертежом требований по точности и шероховатости могут быть применены следующие методы доводочной обработки:

- ручная (слесарная);
- физико-химическая;
- абразивная.

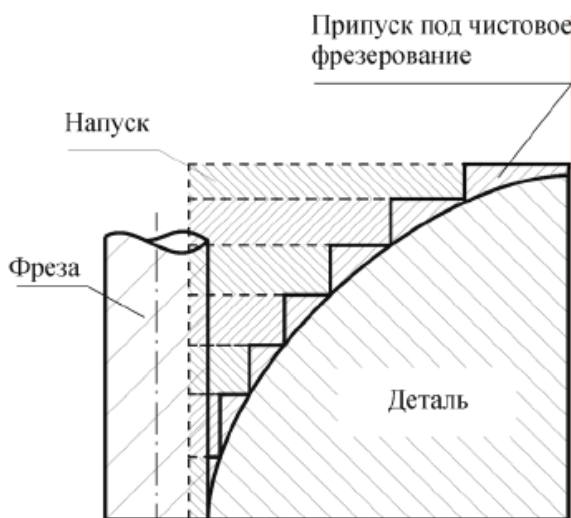


Рисунок 1.1 – Послойное снятие напуска при черновом фрезеровании

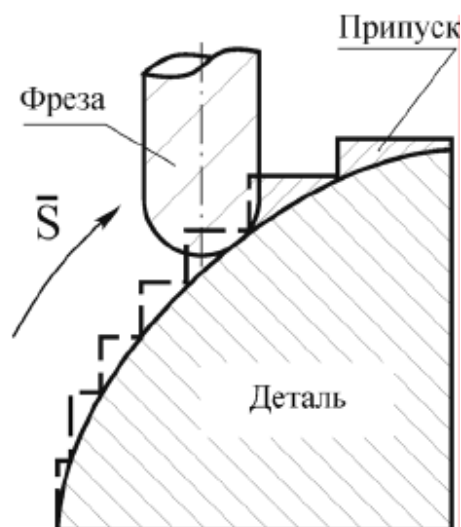


Рисунок 1.2 – Снятие ступенчатого припуска при чистовом фрезеровании

По данным заводов Уральской зоны [3], затраты времени на отделочно-доводочную обработку могут составлять 10...35% от общей трудоемкости обработки и ее продолжительность в 7...10 раз превышает по длительности аналогичные операции чистового фрезерования на станках с ЧПУ. Поэтому целесообразно было бы повысить точность чистового фрезерования, а отделочно-доводочные операции свести к изменению шероховатости, и, как следствие, повысить производительность обработки деталей пространственно-сложной формы.

1.3 Проблемы в обеспечении точности

Погрешность обработки, формируемая на операции фрезерования, может быть выражена в виде векторной суммы [5]:

$$\overline{\Delta_{\Sigma}} = \overline{\Delta_{пн}} + \overline{\Delta_{пр}} + \overline{\Delta_{к}} + \overline{\Delta_{э0}} + \overline{\Delta_{эн}} + \overline{\Delta_{кон}},$$

где $\overline{\Delta_{пн}}$ – погрешность, вносимая промежуточным носителем информации; $\overline{\Delta_{пр}}$ – погрешности, возникающие при преобразовании информации с промежуточного носителя в естественную форму; $\overline{\Delta_{к}}$ – кинематические погрешности способа обработки;

					Лист
Изм.	Лист	№ докум.	Подпись	Дата	15

$\overline{\Delta_{\text{ЭО}}}$, $\overline{\Delta_{\text{ЭН}}}$ – соответственно, обратимые и необратимые погрешности, возникающие от действия энергетических потоков;
 $\overline{\Delta_{\text{КОН}}}$ – погрешности контроля.

Анализ литературных источников [3, 4, 5] показал, что наибольшее влияние на точность обработки при чистовом фрезеровании оказывают энергетические потоки, подводимые или возникающие в процессе обработки. При этом наибольшее влияние оказывает динамическая погрешность от упругих перемещений. В работе [3] отмечается, что она может достигать 50% от суммарной погрешности обработки. Чистовое фрезерование пространственно-сложных поверхностей протекает с непрерывным изменением геометрических параметров зоны резания, которое вызывает силовую нестабильность процесса резания. Нестабильность сил резания, в свою очередь, может вызвать значительные колебания динамических погрешностей и, как следствие, погрешности обработки в пределах одной пространственно-сложной поверхности. Таким образом, существенно повысить точность размеров и формы обрабатываемой фасонной поверхности [3] можно за счет коррекции динамических погрешностей, возникающих от нестабильности действующих сил резания.

Существует два основных метода управления точностью чистового фрезерования:

- изменение режимов резания;
- изменение геометрии зоны резания.

Первый метод главным образом сводится к назначению подачи на зуб с учетом требуемой точности обработки и рассматривается в работах [3, 4, 5].

В работе [4] В.А. Батуевым предложена математическая модель управления подачей на зуб для достижения заданной точности обработки при объемном фрезеровании с учетом работы радиусного участка фрезы. Модель устанавливает функциональную взаимосвязь между подачей на зуб с погрешностью размера динамической настройки, углом наклона

					Лист
					16
Изм.	Лист	№ докум.	Подпись	Дата	

обрабатываемой поверхности и ее конфигурацией, выраженной через положение и длину активной части режущей кромки, физико-механическими свойствами обрабатываемого материала, жесткостью технологической системы и степенью износа зуба фрезы. Данная модель дополнена В.И. Гузеевым [5] с учетом работы цилиндрического участка фрезы. При этом обе модели разработаны для чистового фрезерования в условиях равномерного припуска.

Второй метод реализуется путем назначения траектории движения режущего инструмента и угловой ориентации фрезы относительно пространственно-сложной поверхности и, следовательно, ступенчатого припуска.

В работе [3] В.В. Батуева определена функциональная взаимосвязь изменения толщины срезаемого слоя, положения и длины активной части режущей кромки в каждый момент поворота фрезы с изменением углов наклона обрабатываемой поверхности и величины ступенчатого припуска, разработаны аналитические зависимости для расчета областей допустимых параметров вектора подачи относительно ступенчатого припуска, обеспечивающих заданную точность при наибольшей производительности. Но данная методика сложна для проектирования в условиях дефицита времени и большой номенклатуры выпускаемых изделий.

Таким образом, основная сложность заключается в проектировании операции, занимающем значительную долю времени.

1.4 Постановка цели и задач

Анализ существующих работ показал:

- при изготовлении деталей пространственно-сложной формы после операции чистового фрезерования остается большой и неравномерный припуск, поэтому трудоемкость последующих доводочных (слесарных) операций велика и может достигать 40% от общей трудоемкости;

						Лист
						17
Изм.	Лист	№ докум.	Подпись	Дата		

- повышение производительности обработки детали пространственно-сложной формы возможно за счет повышения точности чистового фрезерования, чтобы оставался минимальный, равномерный припуск, а в функции доводочной операции сводились к улучшению шероховатости;

- отсутствуют нормативные рекомендации по назначению траектории движения режущего инструмента при проектировании операции объемного фрезерования различных конструктивных элементов, составляющих пространственно-сложные поверхности;

- существует два основных метода управления точностью чистового фрезерования: изменение режимов резания и изменение геометрии зоны резания. Оба метода управления точностью обработки можно реализовать в САМ системах при автоматизированном получении управляющих программ. Однако, их реализация характеризуется высокой трудоемкостью.

В связи с этим цель работы – снизить трудоемкость проектирования операций механической обработки поверхностей пространственно-сложной формы.

Для достижения поставленной цели необходимо решить следующие задачи:

- 1) выявить основные конструктивные элементы, из которых формируются пространственно-сложные поверхности;

- 2) провести анализ стратегий обработки, применяемых в САМ системах;

- 3) выбрать расчетные зависимости для определения погрешности размера динамической настройки при чистовом фрезеровании;

- 4) разработать методику проектирования операций механической обработки пространственно-сложных поверхностей;

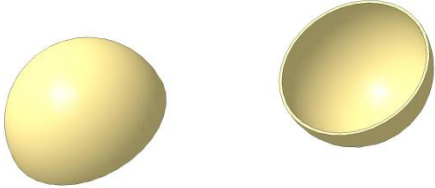


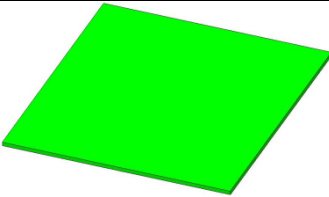
- 5) оценить трудоемкость проектирования операций с применением разработанной методики инженерного проектирования.

						Лист
						18
Изм.	Лист	№ докум.	Подпись	Дата		

2 АНАЛИЗ КОНСТРУКТИВНЫХ ЭЛЕМЕНТОВ, СОСТАВЛЯЮЩИХ ПРОСТРАНСТВЕННО-СЛОЖНЫЕ ПОВЕРХНОСТИ

Исходными данными для автоматизированного проектирования операции механической обработки в САМ системе является САД-модель детали. При проектировании операции механической обработки деталь разбивается на ряд конструктивных элементов, для каждого из которых технологом-программистом назначаются технологические переходы обработки. Таким образом, любую пространственно-сложную поверхность можно представить в виде набора конструктивных элементов. Наиболее распространенные конструктивные элементы, составляющие поверхности пространственно-сложной формы, представлены в таблице 2.1.

Таблица 2.1 – Наиболее распространенные конструктивные элементы
пространственно-сложных поверхностей

Наименование конструктивного элемента	Внешний вид
Сферическая поверхность (выпуклая / вогнутая)	
Цилиндрическая поверхность (выпуклая / вогнутая)	
Коническая поверхность (выпуклая / вогнутая)	
Плоскость	

3 АВТОМАТИЗИРОВАННЫЙ СПОСОБ ПОЛУЧЕНИЯ УПРАВЛЯЮЩИХ ПРОГРАММ

3.1 Описание способа автоматизированного получения управляющих программ

В условиях современного производства подготовка управляющих программ для обработки деталей пространственно-сложной формы осуществляется автоматизированным способом.

В настоящее время основным средством автоматизации подготовки управляющих программ для станков с ЧПУ являются Computer Aided Manufacture (CAM) системы.

Исходными данными для проектирования операции механической обработки в САМ системе является САD-модель детали. При проектировании операции механической обработки в САМ системе деталь разбивается на ряд конструктивных элементов, для каждого из которых технологом-программистом назначаются технологические переходы обработки. Таким образом, технолог-программист формирует траекторию движения инструмента. Выходными данными такой процедуры программирования является фактически массив положений режущего инструмента (CutterLocations DATA, CLDATA), который затем пропускается через постпроцессор, с целью получения управляющей программы. Структура проектирования операции механической обработки в САМ системе представлена на рисунке 3.1.

При разработке управляющей программы, может быть применена одна из стратегий формирования траектории движения инструмента либо их комбинация. Применяя к детали те или иные стратегии формирования траектории, можно обеспечить наиболее благоприятные условия резания и сократить долю вспомогательного времени, связанного с холостыми перемещениями, что окажет влияние на точность и производительность операции фрезерования. Но в тоже время решения, принимаемые на стадии

						Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

проектирования операции механической обработки в САМ системе, зависят от опыта, квалификации, а порой и интуиции технолога, что оказывает существенное влияние на точность и производительность операции в целом.

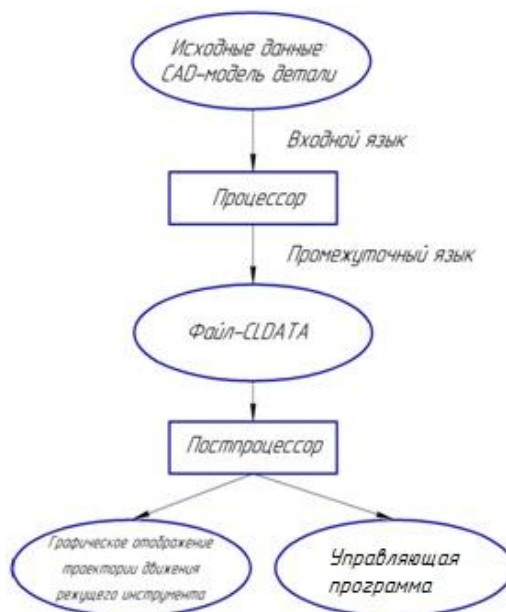


Рисунок 3.1 – Структура проектирования операции механической обработки в САМ системе

3.2 Анализ рынка САМ систем

В настоящее время насчитывается около 60 программных продуктов, относящихся к категории САМ, из них примерно половина имеет международное признание (таблица 3.1) и только два десятка представлены на отечественном рынке. Разработчики САМ систем делятся на два типа: САМ-ориентированные компании, основным продуктом которых является именно САМ, и компании, использующие САМ в составе CAD/CAM и PLM комплексов.

Что касается географии разработки «местных» САМ систем, то безоговорочным лидером является США, далее следует Великобритания, замыкает тройку Россия.

Наиболее распространенными на отечественном рынке (рисунок 3.2) являются следующие САМ продукты:

- PowerMill;
- Mastercam;
- SprutCAM;
- ADEM.

Узнаваемость бренда САМ системы в России

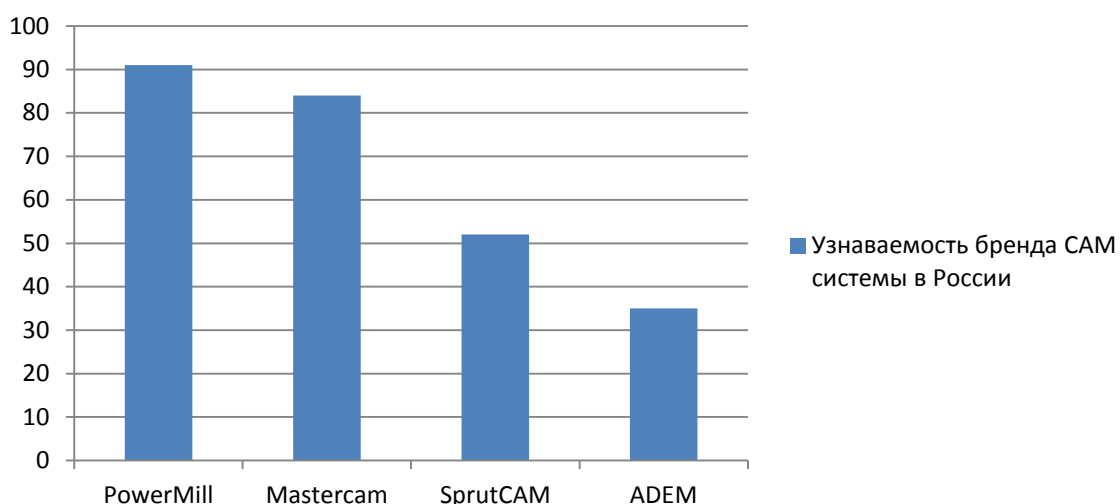


Рисунок 3.2 – Узнаваемость бренда САМ системы в России

Данные о количестве сотрудников, занятых разработкой программного продукта и числе сотрудников местных офисов (без них нормальная эксплуатация программы будет затруднена) приведены на рисунке 3.3.

Таблица 3.1 – Популярные САМ системы и компании-разработчики

САМ систем	компания-разработчик
ADEM	ADEM
Alphacam	Vero Software (HEXAGON)
ArtCAM	Delcam (Autodesk)
BobCAD-CAM	BobCAD-CAM
CAM-TOOL	C&G Systems Inc.
CAMWorks	Geometric Technologies
CATIA	DassaultSystemes
CimatronE	Cimatron Group
Creo (Pro-E)	PTC
Edgecam	Vero Software (HEXAGON)
ESPRIT	DP Technology
FeatureCAM	Delcam (Autodesk)
GeMMA 3D	NTC Gemma
GibbsCAM	Cimatron Group
G02cam	Go2cam International
Inventor HSM/HSMWorks	HSMWorks (Autodesk)
hyperMILL	OPEN MIND
Mastercam	CNC Software
NCGCAM	NCG CAM Solutions
NX	Siemens PLM Software
PartMaker	Delcam (Autodesk)
PEPS	Vero Software (HEXAGON)
PowerMILL	Delcam (Autodesk)
SharpCam	SharpCam Ltd.
SolidCAM/InventorCAM	SolidCAM
Space-E	NTT DATA ENGINEERING SYS
SprutCAM	SPRUT Technologies
SurfCAM	Vero Software (HEXAGON)
Tebis	TebisTechnische Inf. AG
Tehtran	NIP-informatica
T-FLEX	Top Systems
TopSolid	Missler Software
VISI	Vero Software (HEXAGON)
VisualCAD/CAM	MecSoft Corporation
WorkNC	Vero Software (HEXAGON)
ZW3D	ZWSOFT

3.3 Анализ функциональности САМ систем

3.3.1 PowerMill

В системе PowerMill используются следующие стратегии чистовой обработки:

- растровая обработка. Обработка проходами параллельно осям X или Y либо под заданным углом (однонаправленное и двунаправленное резания);
- обработка проходами с постоянной Z. Используется обычно для обработки крутых участков модели;
- обработка пологих участков. Обработка зон, имеющих наклон не более заданного значения. Это достигается технологией использования границ. Эта технология часто используется в сочетании с обработкой с постоянной Z, что дает хорошее качество поверхности по всей модели и оптимальные по времени траектории;
- спиральная обработка. Обычно используется для круглых изделий при высокоскоростной обработке. Для нее характерна низкая нагрузка на инструмент, хорошее качество поверхности и плавность хода;
- радиальная обработка. При этом виде обработки инструмент движется от центра к периферии или наоборот. Наиболее эффективна для изделий типа колец;
- проекционная обработка. Оригинальный способ задания зоны обработки – задается как бы источник света определенной формы, обрабатываться будут те участки модели, которые подсвечены. Возможны 3 варианта:
 - от плоскости – источник света – плоскость, задаются ее положение, размеры и углы наклона и поворота, стратегия – растровая;
 - от линии – источник света – линия, задаются ее положение, размеры и углы обзора. Эффективна для поверхностей типа участков

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		24

цилиндра (в том числе наклонного). Стратегии – линейная, круговая, спиральная;

- от точки – источник света – точка, задаются ее положение и углы обзора в двух плоскостях. Дает постоянный шаг на поверхности типа сферы. Стратегии – круговая, спиральная, радиальная;

- доработка инструментом меньшего радиуса снимает припуск только там, где он остался после обработки большим инструментом. Возможно последовательное применение для нескольких инструментов. Существенно снижает общее время обработки за счет рационального выбора инструмента;

- подчистка углов быстро удаляет оставшийся припуск из острых внутренних углов, при этом шаг проходов определяется высотой гребешка;

- карандашное фрезерование. В этом случае инструмент движется вдоль острых внутренних углов. Поиск этих углов осуществляется автоматически;

- 3D смещение. Эта траектория обеспечивает постоянный шаг по поверхности независимо от ее наклона. Обычно используется совместно с границами. Наиболее экономичная стратегия с точки зрения длины траектории;

- ротационное фрезерование. Это 4-осевая обработка с вращением вокруг оси X. Эффективна для обработки электродов с одного станова. Стратегии – линейная, круговая, винтовая (спиральная).

Примеры проектирования операций чистового фрезерования в системе PowerMill приведены на рисунках 3.6, 3.7.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		25

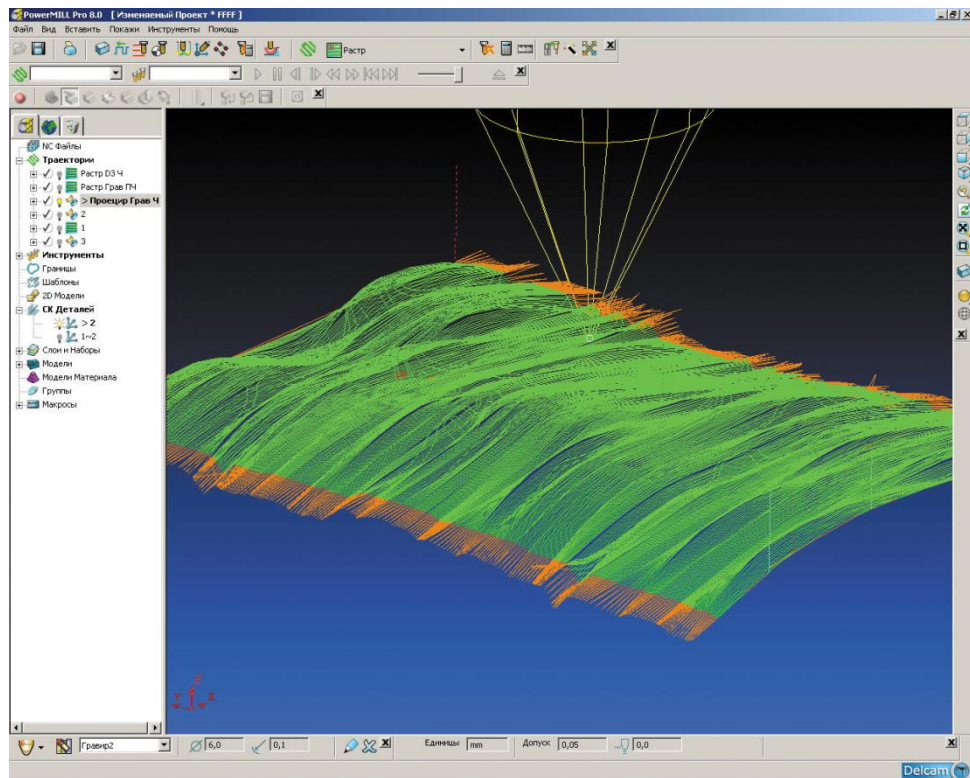
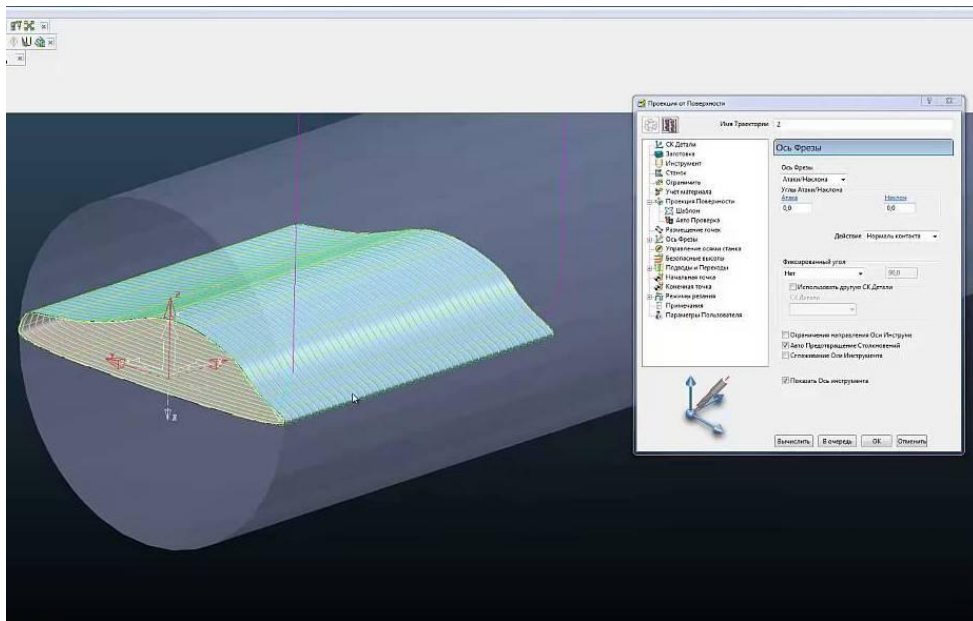


Рисунок 3.6 – Примеры проектирования операций чистового фрезерования в системе PowerMill

Изм.	Лист	№ докум.	Подпись	Дата

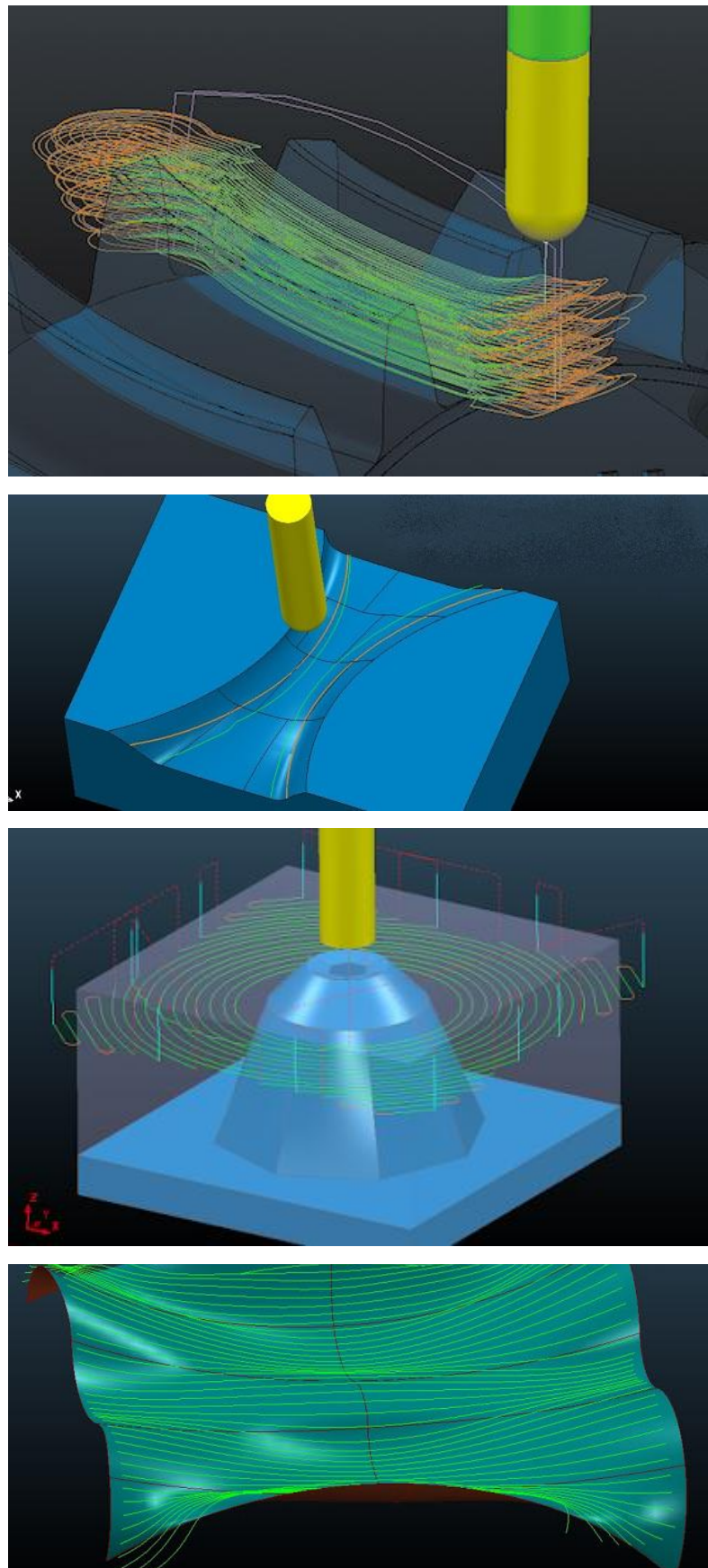


Рисунок 3.7 – Примеры проектирования операций чистового фрезерования в системе PowerMill

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		27

3.3.2 Mastercam

Набор опций Mastercam'a для чистовой обработки включает следующие методы:

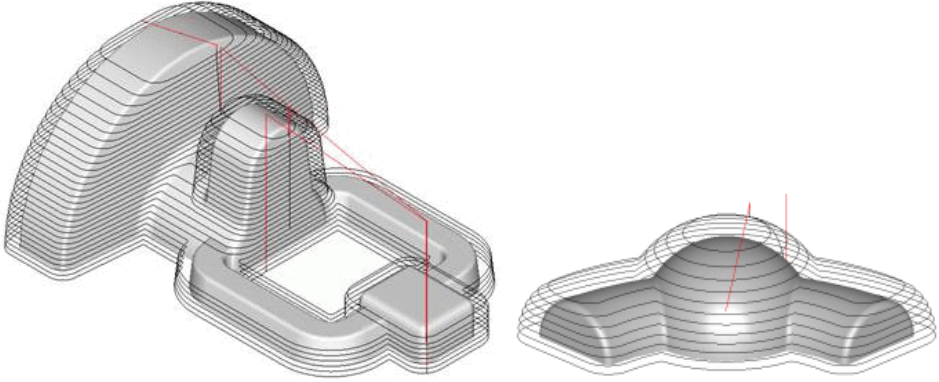
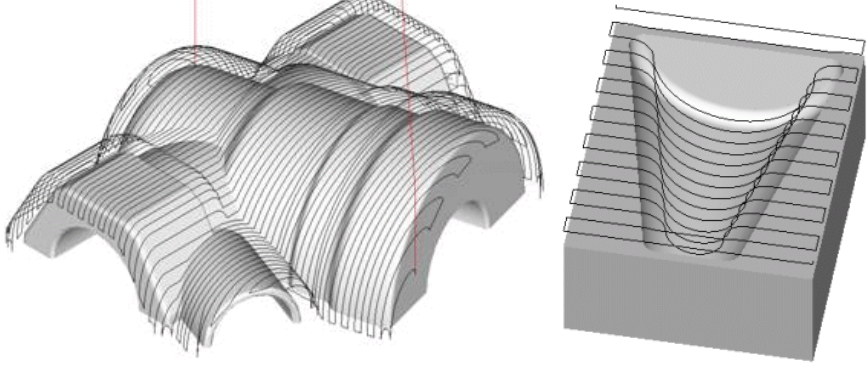
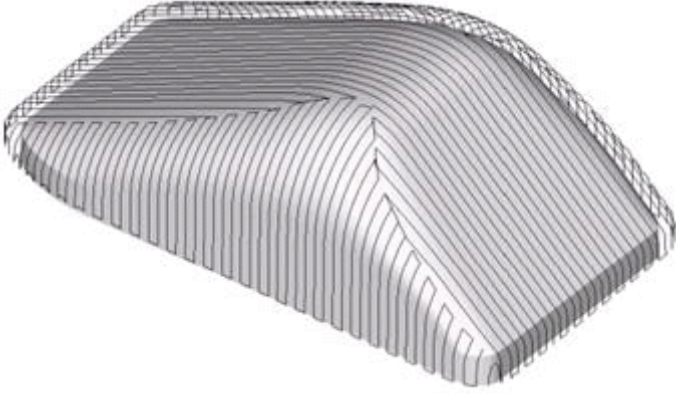
- параллельная чистовая обработка;
- 3D проекционная обработка создает последовательную, гладкую обработку, благодаря отслеживанию кривых многократных поверхностей или тел;
- обработка “scallop” поддерживает такое же качество чистовой обработки на наклонных и плоских поверхностях подобно использованию последовательной, пошаговой 3D обработки;
- радиальная чистовая обработка создает траекторию, ориентированную на внешнюю сторону от выбранной точки, идеально для круглых частей;
- обработка Flowline отдельной или многократных поверхностей, использующая их естественную форму для определения траектории, оставляет более гладкую поверхность;
- Управляемая траектория обработки включает автоматическую синхронизацию телом, элементом или узлом. Опции обработки – постоянный Z, зигзаг, вращение, односторонняя или 5 осевая механическая обработка методом swarf (обработка наклонных поверхностей боковой кромкой фрезы).

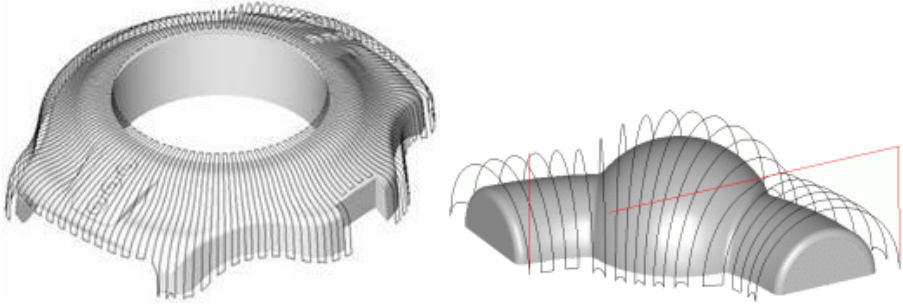
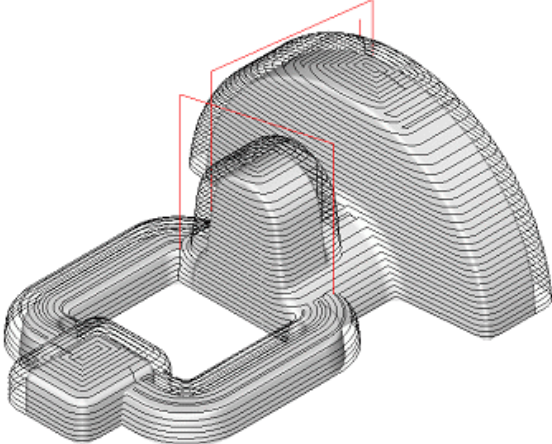
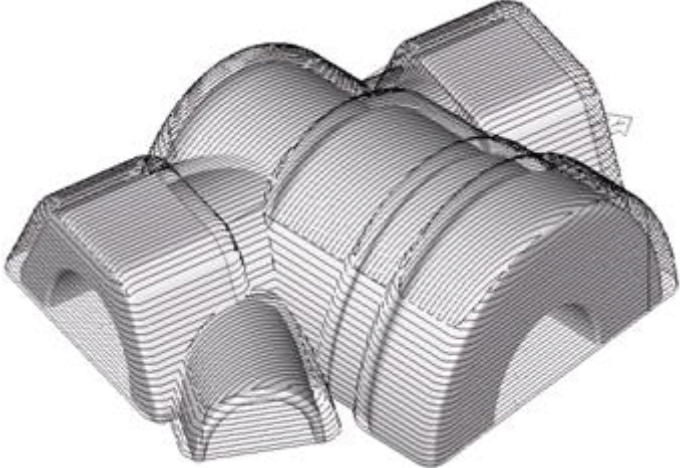
						Лист
Изм.	Лист	№ докум.	Подпись	Дата		28

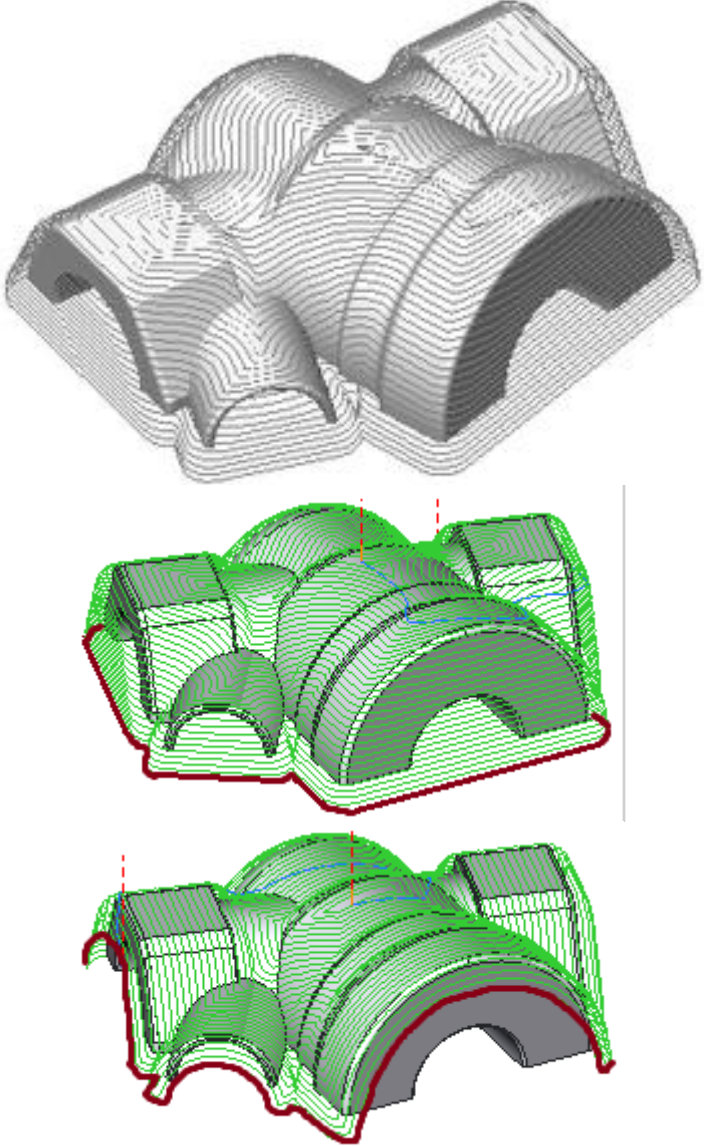
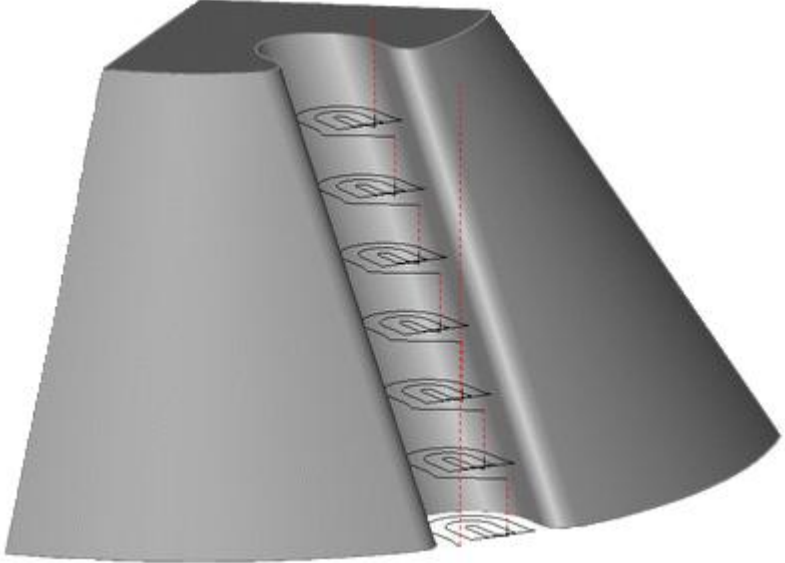
3.3.3 SprutCAM

Для проектирования 3-х координатной обработки в SprutCAM существуют операции, представленные в таблице 3.2.

Таблица 3.2 – Операции 3-х координатной обработки в SprutCAM

№ п/п	Название операции	Рисунок
1	чистовая последняя операция	
2	чистовая построчная операция	
3	чистовая построчная оптимизированная операция (построчно-построчная)	

4	чистовая управляемая операция	
5	чистовая комбинированная операция (последовательно управляемая)	
6	чистовая комплексная операция (последовательно-построчная)	

<p>7</p>	<p>операция «3D смещение»</p>	
<p>8</p>	<p>Операции доработки остаточного материала</p>	

По данным компании «Спрут-Технология», чистовая послойная операция дает хороший результат при обработке деталей или их частей с преобладанием участков поверхности близких к вертикальным. Для последующей обработки пологих участков можно использовать построчную или управляемую чистовую операцию.

Чистовую построчную операцию целесообразно применять для обработки пологих участков поверхности детали, а также участков близких к вертикальным при небольшом отклонении нормали поверхности от плоскости рабочего хода. Для последующей доработки участков с большим наклоном лучше использовать послойную операцию или другую построчную операцию, у которой плоскости строчек перпендикулярны плоскостям строчек первой операции.

Чистовая построчная оптимизированная операция состоит из двух чистовых построчных траекторий, рабочие ходы которых лежат во взаимно перпендикулярных плоскостях. При этом каждая траектория обрабатывает только те зоны, в которых фронтальный угол к поверхности в направлении перемещения инструмента меньше или равен 45 градусам. Это гарантирует, что одни и те же участки не будут обработаны дважды. Кроме того, при такой стратегии достигается равномерная высота гребешка по всей поверхности детали.

Чистовая управляемая операция успешно применяется при обработке отдельных зон детали со сложными вытянутыми криволинейными поверхностями. Целесообразно использовать её для доработки участков поверхности детали специфической формы, для обработки ряда деталей с плавным изменением геометрии поверхности, а так же для фрезерования надписей и рисунков на поверхности детали. При использовании чистовой управляемой операции для обработки пологих участков поверхности детали рекомендуется использовать в качестве направляющих кривых их внешнюю

огибающую и стратегию вдоль кривых. А при обработке крутых участков – стратегию поперёк кривых с такими же направляющими.

Чистовой комбинированной операцией одинаково хорошо обрабатываются как пологие, так и крутые участки. Достаточно равномерная высота гребешка получается даже при использовании постоянного шага. Комбинированная обработка обеспечивает более легкие условия работы инструмента, что в свою очередь позволяет применять удлиненный инструмент меньшего диаметра.

Чистовая комплексная операция состоит из двух траекторий: построчной и послышной. При этом построчная траектория обрабатывает участки, где угол наклона нормали к поверхности меньше угла разделения, а послышная – наоборот, те участки, где угол наклона поверхностей больше угла разделения. Угол разделения по умолчанию составляет 45 градусов. В результате, пологие поверхности обрабатываются построчной стратегией, а более крутые участки – послышной. Преимуществами такого подхода являются: оптимальные режимы резания, равномерная высота гребешка и как следствие сокращенное машинное время.

Траектория операции «3D смещения» начинается с кривых, лежащих на детали, и строится путем последовательно смещения этих кривых вовнутрь на расстояние шага обработки до тех пор, пока кривые не схлопнутся. Таким образом, траектория напоминает эквидистантную выборку за исключением того, что смещение строится не на плоскости, а в 3D пространстве на поверхности детали. В результате, операция позволяет получать постоянную высоту гребешка в независимости от наклона поверхностей обработки. Операция хорошо подходит для получистовой и чистовой обработки сложных деталей с минимальным количеством холостых перемещений при соблюдении попутного или встречного типа фрезерования.

На текущий момент доступны два способа определения начальных кривых:

						Лист
						33
Изм.	Лист	№ докум.	Подпись	Дата		

- начинать с нижнего уровня;
- начинать с верха вертикальных стенок.

В первом случае траектория начинается на нижнем уровне обработки, во втором – с кривых силуэта (проекции обрабатываемы поверхностей на горизонтальную плоскость). Траектория может быть сформирована как изнутри-наружу так и снаружи-внутри. Реализована возможность обработки по спирали вместо параллельных проходов для уменьшения количества переходов. Доступно скругление острых углов в траектории для высокоскоростной обработки.

Преимущества:

- один заход и один выход. Всю деталь возможно обработать при помощи одной непрерывной спиралевидной траектории всего с одним заходом и одним выходом;
- высокоскоростная обработка. Возможна генерация траектории с закругленными углами.

Операции доработки позволяют производить обработку только в тех областях, в которых остался не удаленный материал после обработки предыдущими операциями. Существует возможность обработки оставшегося материала после виртуальной обработки произведенной заданным списком инструментов.

Их целесообразно использовать для черновой или чистовой доработки инструментом другой формы или меньшего диаметра, чем у предыдущих операций. Применение доработки позволяет значительно сократить время обработки на станке деталей сложной формы без дополнительных трудозатрат технолога.

Доработка остаточного материала производится теми же самыми операциями, что и основная обработка, только по-другому инициализируются параметры. По умолчанию выбирается инструмент меньшего диаметра и устанавливается заготовка в виде остаточного

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		34

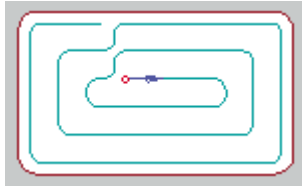
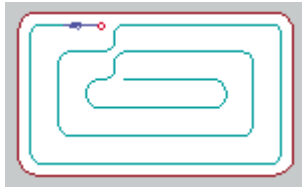
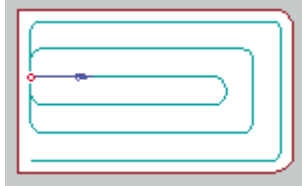
материала. При доработке черновые операции производят выборку всего доступного остаточного материала. А чистовые обрабатывают поверхность детали только в тех местах, где имеется недоработка.

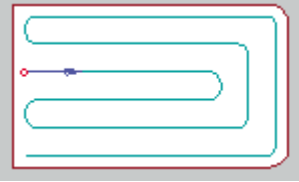
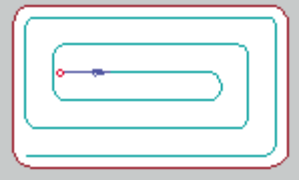
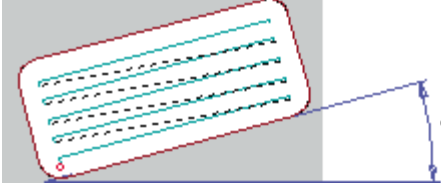
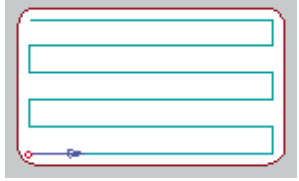
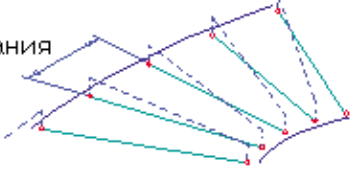
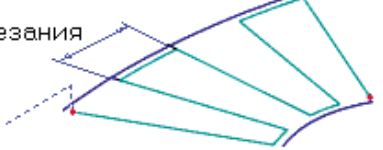
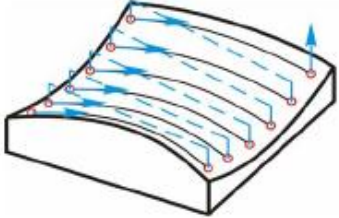
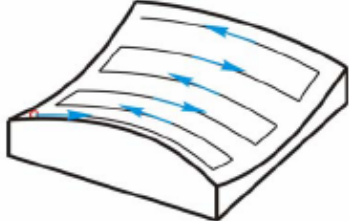
Все технологические особенности операций доработки остаточного материала полностью совпадают с обычными технологическими операциями такого же типа.

3.3.4 ADEM

Независимо от количества управляемых координат, может быть применена одна из стратегий формирования траектории движения инструмента либо их комбинация. Выбор стратегий для той или иной детали оставлен за пользователем. В системе ADEM используются схемы обработки, представленные в таблице 3.3.

Таблица 3.3 – Схемы обработки, используемые в системе ADEM

№ п/п	Название	Рисунок
1	эквидистанта	
2	обратная эквидистанта	
3	петля эквидистантная	

4	зигзаг эквидистантный	
5	спираль	
6	петля	
7	зигзаг	
8	петля контурная	<p data-bbox="900 1088 1126 1122">Глубина резания</p> 
9	зигзаг контурный	<p data-bbox="908 1281 1126 1314">Глубина резания</p> 
10	петля UV	
11	зигзаг UV	

Пример моделирования обработки по спирали и с использованием метода квазиобъемного фрезерования в системе ADEM приведены на рисунках 3.8 и 3.9.

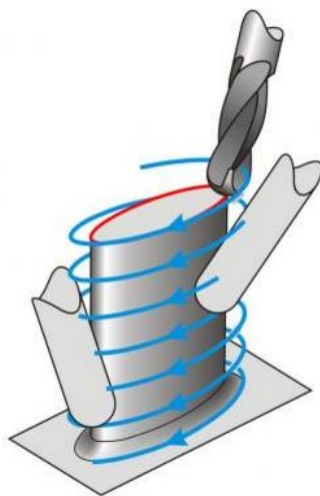


Рисунок 3.8 – Моделирование обработки по спирали в системе ADEM

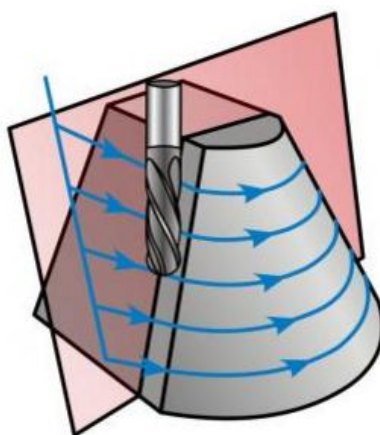


Рисунок 3.9 – Метод квазиобъемного фрезерования в системе ADEM

3.4 Выводы

Обзор наиболее распространенных на отечественном рынке САМ систем позволяет сделать следующий вывод: в различных САМ системах при проектировании операции механической обработки используется одни и те же стратегии.

4 РАЗРАБОТКА ПРОГРАММЫ РАСЧЕТА ПОГРЕШНОСТИ РАЗМЕРА ДИНАМИЧЕСКОЙ НАСТРОЙКИ ПРИ ЧИСТОВОМ ФРЕЗЕРОВАНИИ ПРОСТРАНСТВЕННО-СЛОЖНЫХ ПОВЕРХНОСТЕЙ СО СТУПЕНЧАТЫМ ПРИПУСКОМ

4.1 Постановка задачи, решаемой с помощью программы

Программа, разрабатываемая в рамках данной работы, позволит рассчитать погрешность размера динамической настройки при чистовом фрезеровании пространственно-сложных поверхностей со ступенчатым припуском.

4.2 Формализация поставленной задачи

Ввиду того, что информация, необходимая для решения поставленной задачи, представляется в разных форматах и по разным правилам, выполняется формализация данных, которая заключается в приведении их к одинаковой форме для возможности сопоставления. Результаты формализации задачи, решаемой с помощью разрабатываемой программы, приведены в таблице 4.1.

Таблица 4.1 – Формализация задачи

Единицы измерения	Тип данных	Обозначение в программе	Обозначение	Смысл
Град.	double	teta	Θ	Угол поворота вертикальной плоскости, образующий ступенчатый припуск, относительно направления движения подачи
Град.	double	psi	ψ	Угол поворота зуба фрезы

Продолжение таблицы 4.1

Град.	double	fin	φ_n	центральные углы между осью вращения фрезы и крайними точками пересечения лезвия с припусками
Град.	double	fv1	φ_{v1}	
Град.	double	fv2	φ_{v2}	
Град.	double	fv3	φ_{v3}	
мм	double	R	R	радиус сферической части фрезы
Мм	double	dz	dz	высота ступеньки припуска
Мм	double	dy	dy	ширина ступеньки припуска
	Int (целый)	n	n	порядковый номер плоскости, формирующей ступеньку
мм	double	A	A	расстояние, пройденное фрезой от исходного положения с шагом равным подаче на зуб
Мм	double	aivrh	aivrh	толщина среза (формула) 3.4, при схеме обработки вверх
Мм	double	aivnz	aivnz	толщина среза (формула) 3.4, при схеме обработки вниз
Мм	double	al	al	толщина среза, усеченного вертикальной плоскостью, формирующий ступенчатый список

Продолжение таблицы 4.1

Мм	double	aj	aj	толщина среза, усеченного горизонтальной плоскостью, формирующий ступенчатый список
Мм	double	G	G	расстояние от начала координат до основания первой ступеньки
Град.	double	beta1	β_1	угол сдвига
Град.	double	beta	β	угол между линией среза и направлением действия равнодействующей среза
	double	Ka	K_a	коэффициент утолщения стружки
Град.	double	gammaf	γ_f	передний угол инструмента
	double	C	C	коэффициенты, зависящие от обрабатываемого материала и переднего угла инструмента
	double	XA	X_A	
	double	XV	X_V	
Град.	double	lamda	λ	угол наклона обрабатываемой поверхности к горизонтальной плоскости в направлении подачи

Продолжение таблицы 4.1

Мм/зуб	double	Sz	S_z	подача на зуб
Мм/мин	double	V	V	скорость главного движения резания
Н	double	Px	P_x^Σ	Составляющие силы резания
Н	double	Py	P_y^Σ	
Н	double	Pz	P_z^Σ	
мм	double	Lz	l_z	Величина фаски износа зуба
мм	double		$S_{пер}$	Величина периодической подачи
МПа	double	sigma	σ_i	физико-механические свойства материала
	int	z	z	количество зубьев фрезы
	int	k	k	количество участков активной режущей кромки
Град.	double	omega	ω	Угол наклона касательной к обрабатываемой поверхности
	double	mu	μ	Коэффициент трения
Н	double	рхpop	$P_{поп}^X$	Проекции силы резания на координатные ос станка
Н	double	руpr	$P_{пр}^Y$	
Н	double	pzos	$P_{ос}^Z$	
$*10^{-5}$ МПа	double	E	E	Модуль упругости первого рода

Окончание таблицы 4.1

кг*м ²	double	J _{пр}	J _{пр}	приведенный момент инерции, эквивалентный обобщенному моменту инерции в различных сечениях по длине фрезы.
Н/мм	double	j _{zc}	j _{zc}	Жесткость технологической системы
мм	double	deltaN	ΔN	Погрешность р-ра динамической настройки
мм	double	P	P	Величина сплошного припуска без учета ступенчатого
мм	double	l	l	Вылет фрезы

4.3 Выбор метода решения поставленной задачи

Для расчета погрешности размера динамической настройки необходимо:

1. рассчитать геометрические параметры зоны резания в зависимости от угла наклона пространственно-сложной поверхности для различных стратегий обработки (угол поворота ступенек припуска относительно вектора подачи θ);
2. рассчитать составляющие силы резания с учетом изменяющихся геометрических параметров зоны резания;
3. рассчитать погрешность размера динамической настройки для различных стратегий обработки.

Последовательность расчета погрешности размера динамической настройки:

1. назначаются исходные данные;
2. рассчитываются “пятна контакта” фрезы с припуском, т.е. определяются пределы интегрирования;
3. рассчитываются толщины среза для элементарных участков активной части режущей кромки участвующих в работе;
4. для каждого элементарного участка активной части режущей кромки определяются углы β_1 и β ;
5. для каждого элементарного участка активной части режущей кромки рассчитываются все три составляющие силы резания;
6. рассчитываются проекции составляющих силы резания на координатные оси станка;
7. рассчитывается погрешность размера динамической настройки;

Расчетная схема обработки представлена на рисунке , расчетная схема для определения толщины срезаемого слоя при фрезеровании поверхности со ступенчатым припуском – на рисунке 5. Разложение сил, действующих на зуб концевой радиусной фрезы, в системе координат станка изображено на рисунке 4.1.

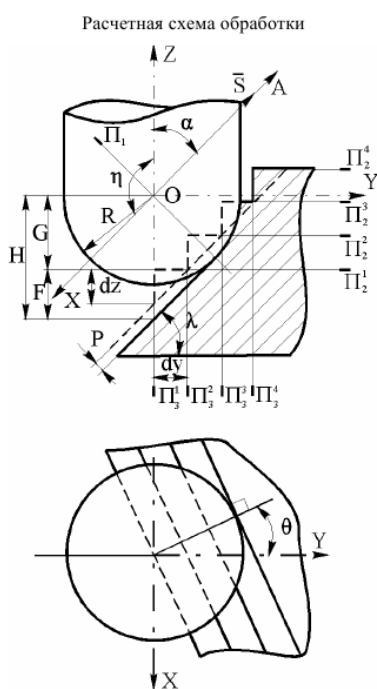


Рисунок 4.1 – Расчетная схема обработки

Расчетная схема для определения толщины срезаемого слоя при фрезеровании поверхности со ступенчатым припуском

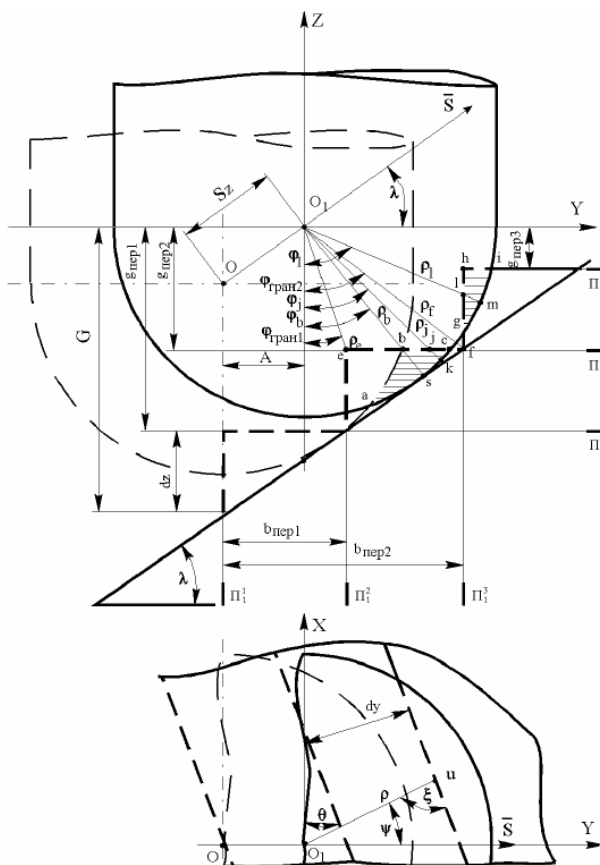


Рисунок 4.2 – Расчетная схема для определения толщины срезаемого слоя при фрезеровании поверхности со ступенчатым припуском

Разложение сил (в системе координат станка), действующих на зуб
концевой радиусной фрезы

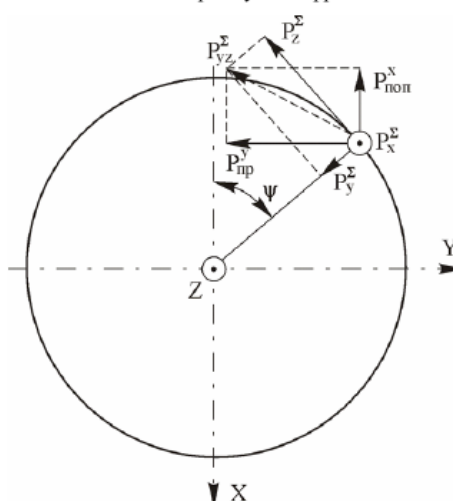


Рисунок 4.3 – Разложение сил, действующих на зуб концевой радиусной фрезы, в системе координат станка

4.4 Разработка алгоритма

Разработанный алгоритм расчета погрешности размера динамической настройки при чистовом фрезеровании пространственно-сложных поверхностей со ступенчатым припуском представлен на рисунках 4.4, 4.5.

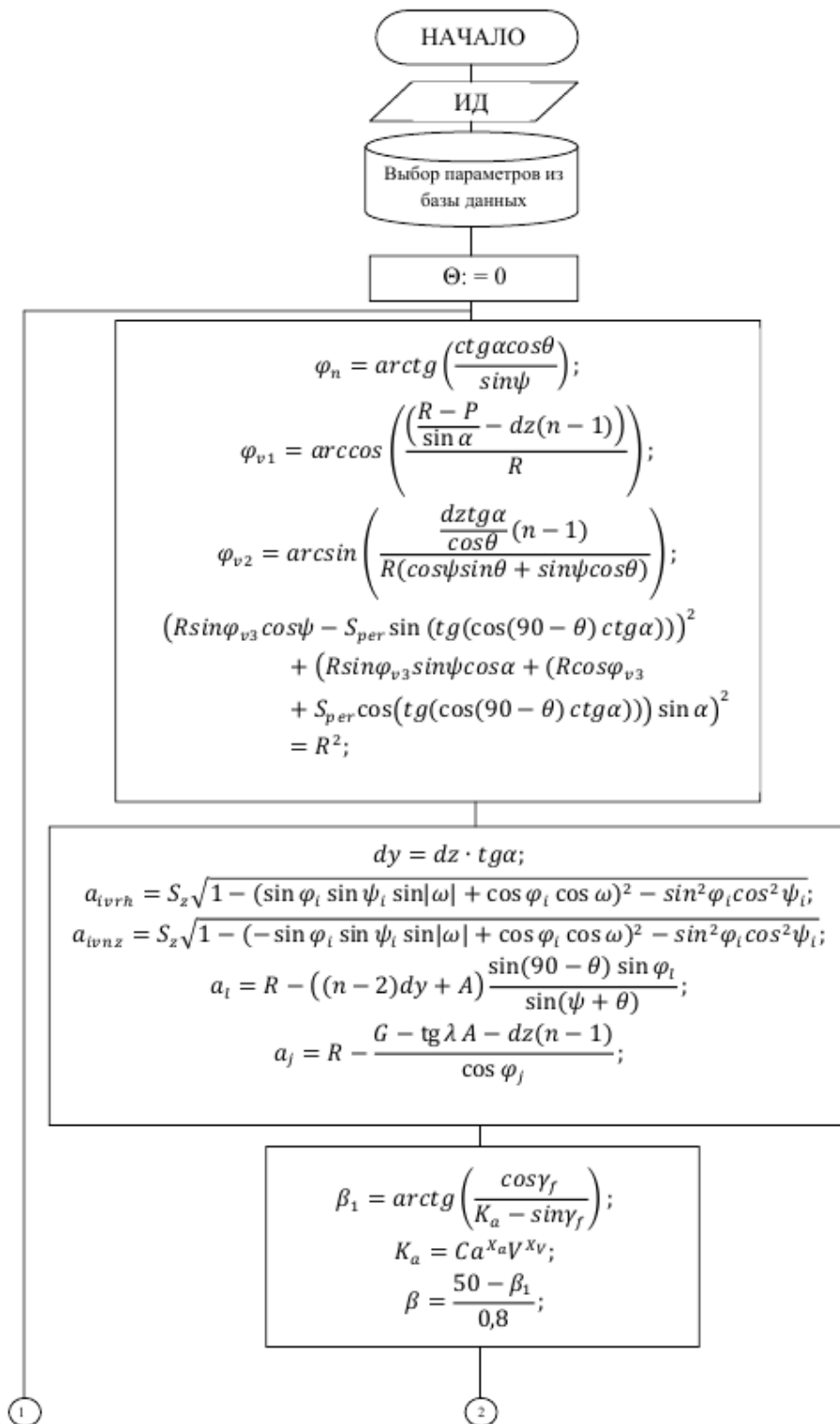


Рисунок 4.4 – Алгоритм программы (начало)

Изм.	Лист	№ докум.	Подпись	Дата

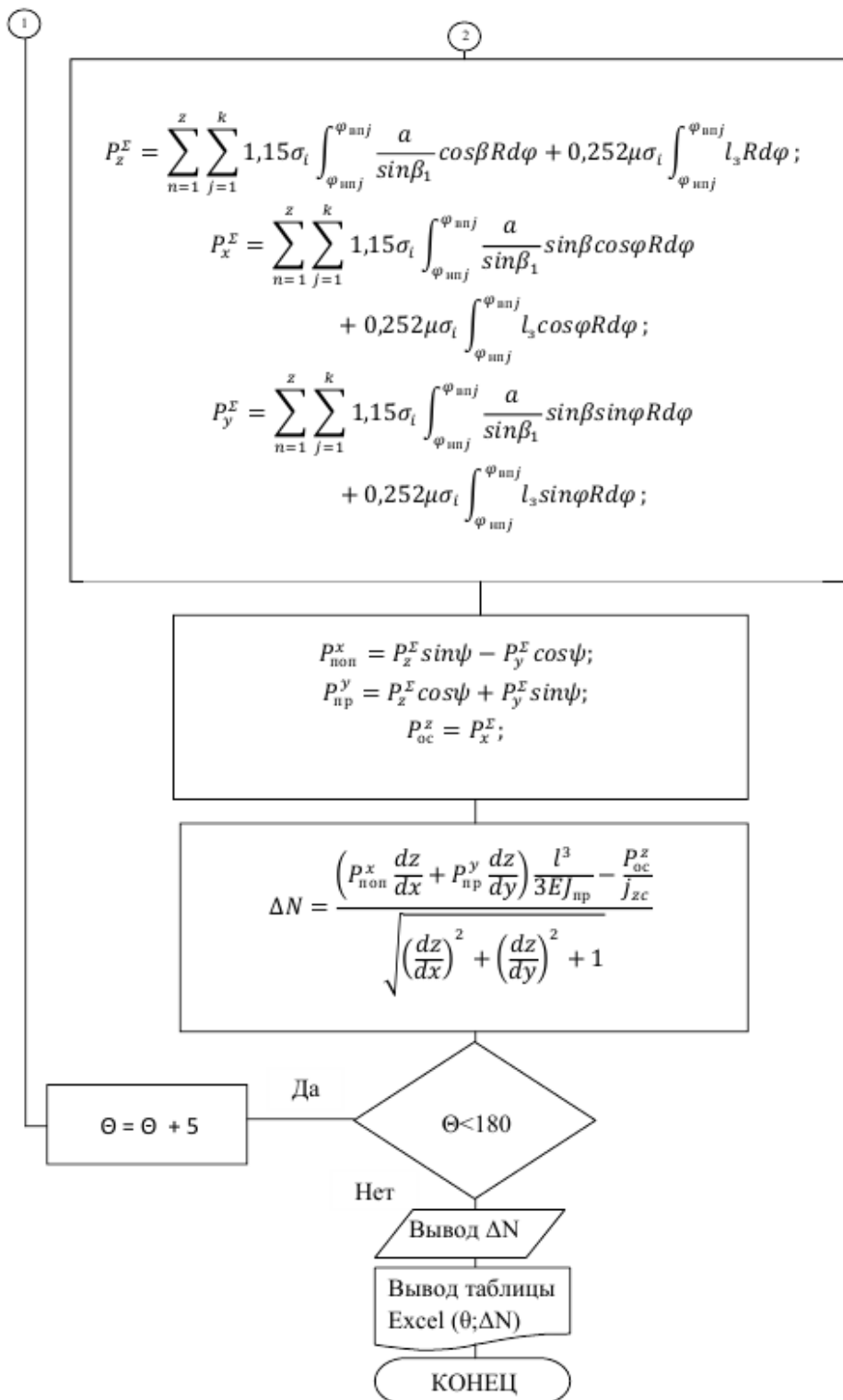


Рисунок 4.5 – Алгоритм программы (конец)

4.5 Написание и отладка программы

По разработанному алгоритму на языке программирования C# написана программа (код программы представлен в приложении А, окно программы представлено на рисунке 4.6).

The screenshot shows a software application window titled "Расчет сил резания" (Calculation of cutting forces). The interface is divided into several sections:

- Выберите тип фрезерования** (Select milling type): Radio buttons for "по следу" (selected) and "по строке" (selected).
- Выберите вид припуска** (Select allowance type): Radio buttons for "равномерный" (selected) and "ступенчатый" (selected).
- Ввод исходных данных:** Input fields for various parameters:
 - $\sigma_f = 1000$ - интенсивность напряжения
 - $S_z = 0,2$ - подача на зуб, мм
 - $I_z = 1$ - износ по задней поверхности, мм
 - $R = 8$ - радиус фрезы, мм
 - $\mu = 1$ - коэффициент трения
 - $dz = 1$ - размер ступени
 - $A = 1$ - смещение инструмента
 - $\theta = 5$ - направление подачи
 - $P = -0,225$ - величина сплошного припуска без учета ступенчатого
 - $\psi = 90$ - угол контакта, град
 - $\beta_1 = 30$ - угол заострения, град
 - $\omega = 30$ - угол наклона обрабатываемой поверхности в горизонтальной плоскости в направлении подачи, град
 - $\omega(n) = 30$ - угол наклона обрабатываемой поверхности в горизонтальной плоскости в направлении перпендикулярном подаче, град
 - $S_{пер} = 4$ - величина периодической подачи
 - $l = 10$ - вылет фрезы, мм
 - $J_{пр} = 1000$ - момент инерции, $kJz_{с} = 300$ - жёсткость, Н/мм
 - $E = 250$ - модуль упругости, МПа
- Buttons:** "Рассчитать" (Calculate) and "Очистить" (Clear).
- Results Section:**
 - $\phi(n) = 29,811^\circ$
 - $R_x = 350,384Н$, $R_{x_поп} = 435,809Н$
 - $R_y = 235,921Н$, $R_{y_пр} = 235,921Н$
 - $R_z = 435,809Н$, $R_{z_ос} = 350,384Н$, $\Delta N = -0,504145213$

Рисунок 4.6 – Программа расчета погрешности размера динамической настройки

4.6 Выводы по разделу

Разработанная в рамках данной работы программа позволит рассчитать погрешность размера динамической настройки при различных параметрах обработки основных конструктивных элементов. Эти расчеты позволят разработать рекомендации по назначению стратегий обработки отдельных конструктивных элементов, на которые разбивается деталь в процессе проектирования операции механической обработки. На основе данных рекомендаций может быть разработана методика инженерного проектирования. В свою очередь, данная методика назначения стратегий

обработки может быть применена при проектировании операции механической обработки пространственно-сложных поверхностей в САМ системе. Это упростит принятие решений технологом при проектировании операции механической обработки пространственно-сложных поверхностей со ступенчатым припуском путем назначения обоснованных стратегий обработки для отдельных конструктивных элементов, составляющих пространственно-сложную поверхность.

						Лист
						48
Изм.	Лист	№ докум.	Подпись	Дата		

5 ОБРАБОТКА РЕЗУЛЬТАТОВ ВКР

5.1 Разработка методики инженерного проектирования

На основе изложенного выше материала разработана методика определения ожидаемых погрешностей размера динамической настройки при различных дискретных углах наклона обрабатываемой поверхности. Данная методика состоит из следующих этапов:

1. Разбиение пространственно-сложной поверхности на элементарные геометрические элементы.
2. Назначение в САМ системе рекомендованных видов и параметров стратегий для каждого элементарного геометрического элемента с учетом заданной точности.
3. Назначение последовательности обработки элементарных геометрических элементов.

Для реализации второго этапа методики выполняется расчет погрешности размера динамической настройки при разных траекториях движения режущего инструмента для каждого конструктивного элемента пространственно-сложной поверхности. Результаты расчетов отражены в виде таблиц (таблицы 5.1 – 5.4) и графиков (рисунки 5.1 – 5.3).

Таблица 5.1 – Результаты расчета погрешности размера динамической настройки при обработке плоских поверхностей

$\omega=30^\circ, \omega_{\text{п}}=0^\circ$							
$\Theta, ^\circ$	5	10	15	20	25	30	35
$\Delta N, \text{мкм}$	170	120	84	73	51	31	25
$\omega=30^\circ, \omega_{\text{п}}=15^\circ$							
$\Theta, ^\circ$	5	10	15	20	25	30	35
$\Delta N, \text{мкм}$	224	158	111	95	66	38	33
$\omega=30^\circ, \omega_{\text{п}}=30^\circ$							
$\Theta, ^\circ$	5	10	15	20	25	30	35
$\Delta N, \text{мкм}$	268	189	132	112	79	43	385

$\omega=30^\circ, \omega_{\text{п}}=45^\circ$							
$\Theta, ^\circ$	5	10	15	20	25	30	35
$\Delta N, \text{мкм}$	298	211	147	124	88	46	424
$\omega=30^\circ, \omega_{\text{п}}=60^\circ$							
$\Theta, ^\circ$	5	10	15	20	25	30	35
$\Delta N, \text{мкм}$	309	219	153	128	91	46	435
$\omega=30^\circ, \omega_{\text{п}}=75^\circ$							
$\Theta, ^\circ$	5	10	15	20	25	30	35
$\Delta N, \text{мкм}$	298	210	147	122	87	42	413
$\omega=45^\circ, \omega_{\text{п}}=15^\circ$							
$\Theta, ^\circ$	5	10	15	20	25	30	35
$\Delta N, \text{мкм}$	192	171	62	13	290	120	75
$\omega=45^\circ, \omega_{\text{п}}=30^\circ$							
$\Theta, ^\circ$	5	10	15	20	25	30	35
$\Delta N, \text{мкм}$	218	194	71	15	329	136	84
$\omega=45^\circ, \omega_{\text{п}}=45^\circ$							
$\Theta, ^\circ$	5	10	15	20	25	30	35
$\Delta N, \text{мкм}$	238	211	77	16	358	147	90
$\omega=45^\circ, \omega_{\text{п}}=60^\circ$							
$\Theta, ^\circ$	5	10	15	20	25	30	35
$\Delta N, \text{мкм}$	244	218	80	16	367	151	92
$\omega=45^\circ, \omega_{\text{п}}=75^\circ$							
$\Theta, ^\circ$	5	10	15	20	25	30	35
$\Delta N, \text{мкм}$	229	204	75	16	344	141	85

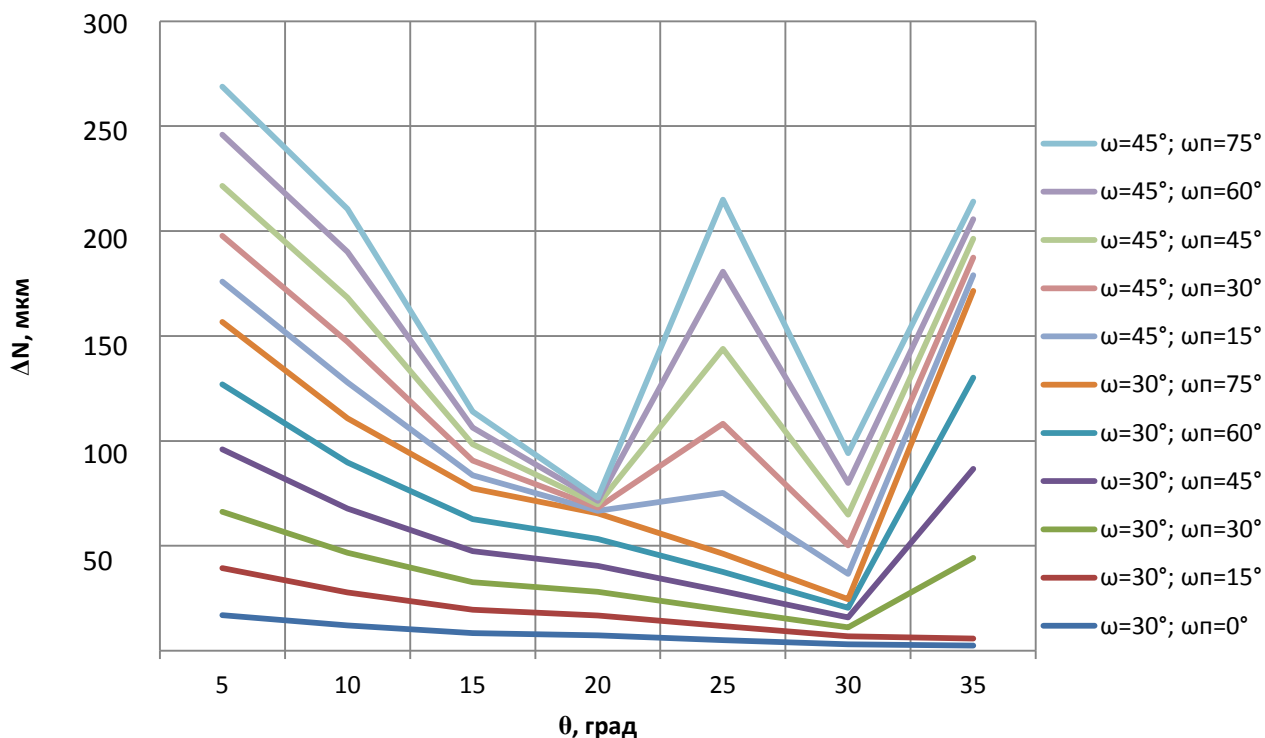


Рисунок 5.1 – Результаты расчета погрешности размера динамической настройки при обработке плоских поверхностей

Таблица 5.2 – Результаты расчета погрешности размера динамической настройки при обработке цилиндрических поверхностей

$\omega_p=30^\circ$							
$\theta,^\circ \backslash \omega,^\circ$	5	10	15	20	25	30	35
15	224	158	111	95	66	38	328
30	268	189	132	112	79	43	386
45	298	211	147	124	88	46	424
$\omega_p=45^\circ$							
$\theta,^\circ \backslash \omega,^\circ$	5	10	15	20	25	30	35
30	298	211	147	124	88	46	424
45	238	211	77	16	358	147	90
75	229	204	75	16	344	141	85
$\omega_p=60^\circ$							
$\theta,^\circ \backslash \omega,^\circ$	5	10	15	20	25	30	35
30	310	219	153	128	91	46	435
45	244	218	80	17	367	151	91

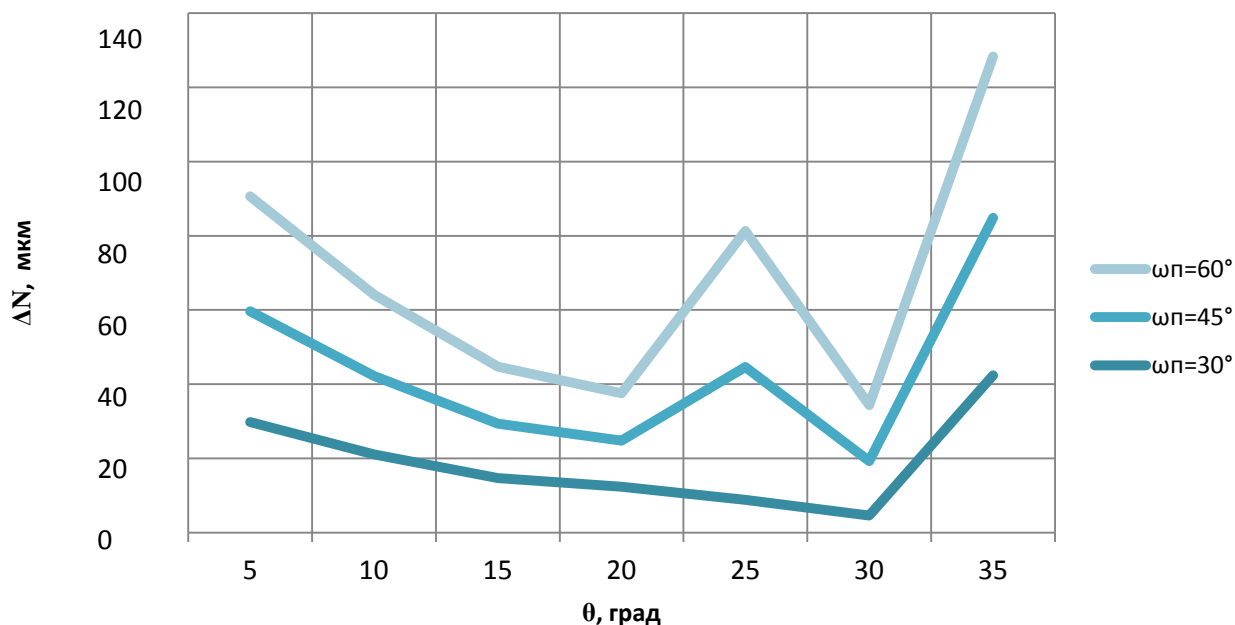


Рисунок 5.2 – Результаты расчета погрешности размера динамической настройки при обработке цилиндрических поверхностей

Таблица 5.3 – Результаты расчета погрешности размера динамической настройки при обработке конических поверхностей

Конусность 1:50 ($\omega_{п}=32,7^{\circ}$)							
$\theta, ^{\circ}$ $\omega, ^{\circ}$	5	10	15	20	25	30	35
30	274	194	135	115	81	44	394
45	223	198	72	15	335	138	85
Конусность 1:10 ($\omega_{п}=16,5^{\circ}$)							
$\theta, ^{\circ}$ $\omega, ^{\circ}$	5	10	15	20	25	30	35
30	229	162	113	97	68	38	334
45	195	173	63	13	294	122	76
Конусность 1:5 ($\omega_{п}=75,3^{\circ}$)							
$\theta, ^{\circ}$ $\omega, ^{\circ}$	5	10	15	20	25	30	35
30	297	210	147	122	87	42	412
45	229	204	75	16	343	141	85

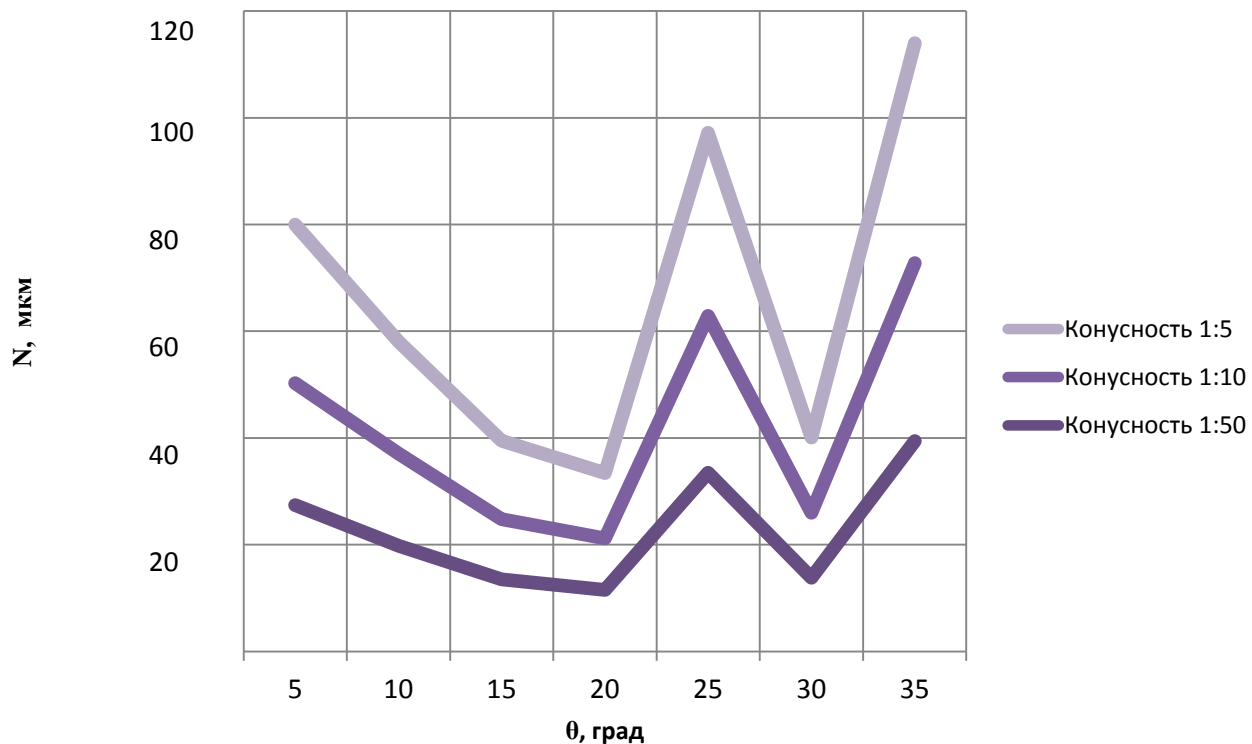


Рисунок 5.3 – Результаты расчета погрешности размера динамической настройки при обработке конических поверхностей

Таблица 5.4 – Результаты расчета погрешности размера динамической настройки при обработке сферических поверхностей

$\Theta = 5^\circ$				
ω_p, \circ ω, \circ	30	45	60	75
30	268	298	310	298
45	218	238	244	229
$\Theta = 10^\circ$				
ω_p, \circ ω, \circ	30	45	60	75
30	189	211	219	210
45	194	211	218	204

$\Theta = 15^\circ$				
$\omega_{п},^\circ$ $\omega,^\circ$	30	45	60	75
30	132	147	153	147
45	71	77	80	75
$\Theta = 20^\circ$				
$\omega_{п},^\circ$ $\omega,^\circ$	30	45	60	75
30	112	124	128	122
45	15	16	17	16
$\Theta = 25^\circ$				
$\omega_{п},^\circ$ $\omega,^\circ$	30	45	60	75
30	79	88	91	87
45	329	357	367	344
$\Theta = 30^\circ$				
$\omega_{п},^\circ$ $\omega,^\circ$	30	45	60	75
30	43	46	46	42
45	136	147	151	141
$\Theta = 35^\circ$				
$\omega_{п},^\circ$ $\omega,^\circ$	30	45	60	75
30	386	424	435	413
45	84	90	92	85

Анализ результата расчетов погрешности размера динамической настройки при обработке сферических поверхностей позволяет сделать вывод о том, что при угле наклона касательной к обрабатываемой сферической поверхности от 45° целесообразно применять стратегию обработки, характеризующуюся углом $\Theta = 20^\circ$, а при угле наклона касательной к обрабатываемой сферической поверхности от $45^\circ - \Theta = 20^\circ$. То есть сферическую поверхность следует дополнительно разбивать на участки, которым соответствуют стратегии обработки с разными траекториями.

На основании выполненных выше расчетов разработаны рекомендации по назначению стратегии обработки, определяющейся углом θ между направлением подачи и горизонтальной проекции границы ступеньки припуска. Данные рекомендации приведены в таблицах 5.5 – 5.8.

Таблица 5.5 – Рекомендации по назначению стратегий обработки плоских поверхностей

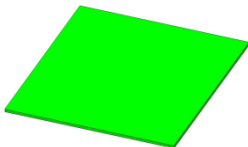
	$\Theta, ^\circ$ – угол между направлением подачи и горизонтальной проекции границы ступеньки припуска					
		15	30	45	60	75
$\omega, ^\circ$ \ / $\omega_p, ^\circ$	30	35	30	30	30	30
	45	20	20	20	20	20

Таблица 5.6 – Рекомендации по назначению стратегий обработки
цилиндрических поверхностей

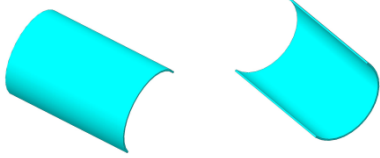
		$\Theta, ^\circ$ – угол между направлением подачи и горизонтальной проекции границы ступеньки припуска		
$\omega, ^\circ$ \ / \ $\omega_{п}, ^\circ$		30	45	60
	≤ 45	30	30	30
	> 45	20	20	20

Таблица 5.7 – Рекомендации по назначению стратегий обработки
конических поверхностей

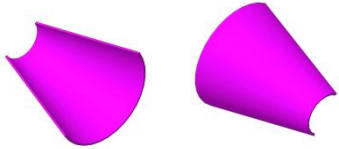
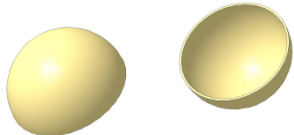
		$\Theta, ^\circ$ – угол между направлением подачи и горизонтальной проекции границы ступеньки припуска		
$\omega, ^\circ$ \ / \ Конусность		1:50	1:10	1:5
	≤ 45	30	30	30
	> 45	20	20	20

Таблица 5.8 – Рекомендации по назначению стратегий обработки
сферических поверхностей

		$\Theta, ^\circ$ – угол между направлением подачи и горизонтальной проекции границы ступеньки припуска		
$\omega, ^\circ$ \ / \ $\omega_{п}, ^\circ$		30	45	60
	≤ 45	30	30	30
	> 45	20	20	20

5.2 Пример применения методики инженерного проектирования

В качестве обработки результатов работы выполним отработку методики на примере одной детали. Реализация первого этапа методики представлена на рисунке 5.4.

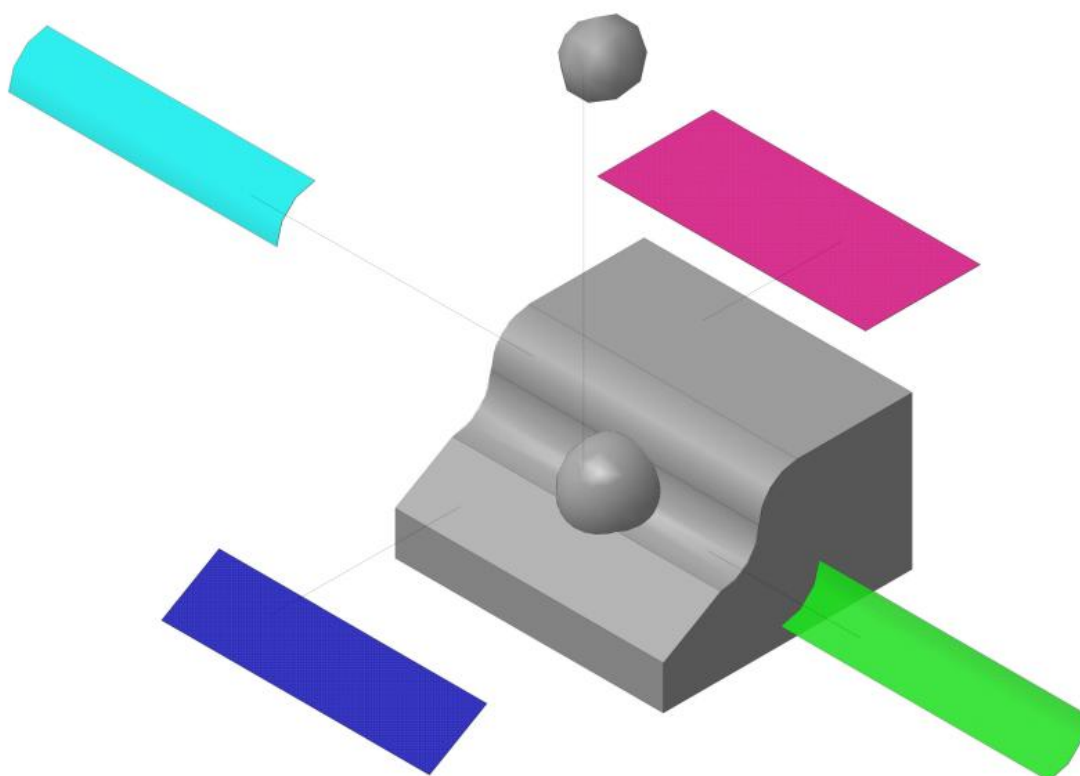


Рисунок 5.4 – Реализация первого этапа методики (разбиение пространственно-сложной поверхности на элементарные геометрические элементы)

По результатам расчетов назначается оптимальная траектория движения режущего инструмента, соответствующая минимальному значению погрешности размера динамической настройки. Таким образом, реализуется второй этап методики инженерного проектирования (таблица 5.9).

Таблица 5.9 – Реализация второго этапа методики (назначение оптимальных траекторий движения режущего инструмента для отдельных конструктивных элементов)

Конструктивный элемент пространственно-сложной поверхности	Угол наклона, соответствующий оптимальной траектории режущего инструмента
Наклонная плоскость	20
Цилиндрическая (вогнутая)	20
Сферическая (выпуклая)	30
	20
Цилиндрическая (выпуклая)	30
Наклонная плоскость	35

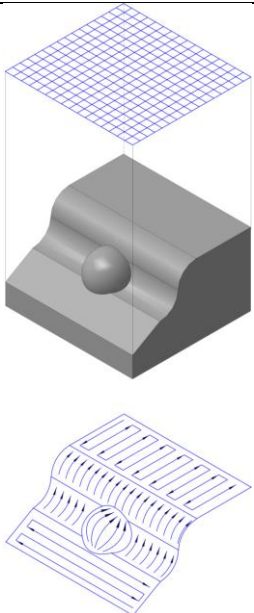
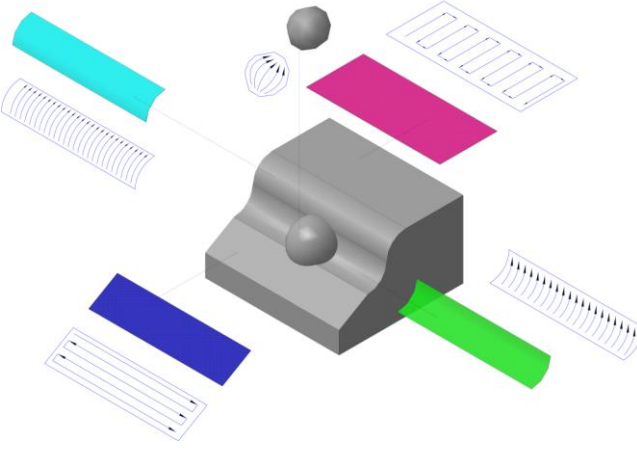
Реализация третьего этапа методики заключается в назначении последовательности обработки элементарных геометрических элементов. Для рассматриваемой детали назначена следующая последовательность обработки конструктивных элементов:

1. наклонная плоскость;
2. сферическая выпуклая;
3. цилиндрическая вогнутая;
4. цилиндрическая выпуклая;
5. наклонная плоскость.

5.3 Оценка трудоемкости проектирования операции механической обработки по разработанной методике

Для оценки трудоемкости проектирования операции механической обработки по разработанной методике сравним ее с трудоемкостью, соответствующей проектированию по старой методике (таблица 5.10).

Таблица 5.10 – Сравнение трудоемкости проектирования с использованием старой и разработанной методик

Старая методика	Разработанная методика
	
<p>Время проектирования операции</p> <p style="text-align: center;">72,9 мин</p>	<p>Время проектирования операции</p> <p style="text-align: center;">48,3 мин</p>

Сравнительный анализ трудоемкости проектирования операций с использованием старой и новой методик показывает, что реализация предложенной методики позволяет снизить трудоемкость проектирования операции механической обработки пространственно-сложных поверхностей в среднем на 40% при обеспечении простоты и надежности проектных решений.

ЗАКЛЮЧЕНИЕ

Современное машиностроение характеризуется постоянным ростом номенклатуры деталей пространственно-сложной формы, к которым предъявляются высокие требования по точности. Большое значение для достижения заданной точности таких деталей имеет стадия чистового фрезерования, на которой происходит снятие ступенчатого припуска и получение точности размеров и формы.

Операции чистового фрезерования пространственно-сложных поверхностей, как правило, выполняются на станках с ЧПУ и проектируются при помощи САМ систем. Однако, несмотря на использование систем автоматизированного проектирования, процесс проектирования подобных операций характеризуется высокой трудоемкостью. В связи с этим, целью работы являлось снижение трудоемкости проектирования операций механической обработки поверхностей пространственно-сложной формы.

При рассмотрении вопроса обработки пространственно-сложных поверхностей на станках с ЧПУ было выявлено два основных метода управления точностью чистового фрезерования пространственно-сложных поверхностей:

- изменение режимов резания;
- изменение геометрии зоны резания.

В рамках данной работы исследован второй метод, связанный с назначением траектории движения режущего инструмента относительно пространственно-сложной поверхности и ступенчатого припуска.

Ввиду отсутствия рекомендаций по назначению стратегий обработки различных пространственно-сложных поверхностей, технолог-программист на стадии проектирования операции принимает решения, основанные на собственном, опыте, квалификации и интуиции. Это, в свою очередь, обуславливает высокую трудоемкость проектирования операций

						Лист
						60
Изм.	Лист	№ докум.	Подпись	Дата		

механической обработки поверхностей пространственно-сложной формы, являющуюся основным фактором актуальности данной работы.

В качестве основного фактора, оказывающего влияние на точность обработки, рассмотрена погрешность размера динамической настройки, возникающая вследствие нестабильности сил резания.

В ходе выполнения работы разработана математическая модель (программа) расчета погрешности размера динамической настройки при разных стратегиях обработки. Использование данной программы позволило разработать рекомендации для некоторых конструктивных элементов, формирующих пространственно-сложные поверхности.

На основе проведенных исследований разработана методика инженерного проектирования, основанная на определении ожидаемых погрешностей размера динамической настройки при различных дискретных углах наклона обрабатываемой поверхности. Данная методика состоит из следующих этапов:

1. Разбиение пространственно-сложной поверхности на элементарные геометрические элементы.

2. Назначение в САМ системе рекомендованных видов и параметров стратегий для каждого элементарного геометрического элемента с учетом заданной точности.

3. Назначение последовательности обработки элементарных геометрических элементов.

Применение предложенной методики позволяет снизить трудоемкость проектирования операции механической обработки пространственно-сложных поверхностей в среднем на 40%.

По результатам работы в рамках проекта ТЕМПУС выполнен доклад на 5-й Международной научно-практической конференции «Современное машиностроение: наука и образование» (секция «Реализация международной сетевой магистерской программы») и опубликована статья (приложение Б).

						Лист
						61
Изм.	Лист	№ докум.	Подпись	Дата		

В качестве направления дальнейших исследований рассматривается расширение библиотеки конструктивных элементов; разработка рекомендаций для большего набора поверхностей.

						Лист
						62
Изм.	Лист	№ докум.	Подпись	Дата		

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кулыгин, В.Л. Основы технологии машиностроения: учебное пособие / В.Л. Кулыгин, И.А. Кулыгина – Челябинск: Издательский центр ЮУрГУ, 2010. – 147 с.;
2. Кулыгин, В.Л. Технология машиностроения: учебное пособие для студентов вузов / В.Л. Кулыгин, В.И. Гузеев, И.А. Кулыгина – Москва: Издательский Дом «БАСТЕТ», 2011. – 184 с.;
3. Батуев, В.В. Повышение производительности и точности чистового фрезерования пространственно-сложных поверхностей со ступенчатым припуском: дис. канд. техн. наук / В.В. Батуев. – Челябинск: Южно-Уральский государственный университет, 2007. – 209 с.;
4. Батуев, В.А. Повышение производительности и точности фрезерования пространственно-сложных поверхностей на станках с ЧПУ путем стабилизации сил резания: дис. канд. техн. наук / В.А. Батуев. – Саратов: Саратовский политехнический институт, 1986. – 248 с.;
5. Гузеев, В.И. Теория и методика производительности контурной обработки деталей разной точности на токарных и фрезерных станках с ЧПУ: дис. докт. техн. наук / В.И. Гузеев. – Челябинск: ЧГТУ, 1994. – 517 с.;
6. Шеховцева, Т.В. Целесообразность применения станков с ЧПУ в единичном и мелкосерийном производстве на основе показателей технологичности деталей ГТД / Т.В. Шеховцева // Вестнике МГТУ, том 14, №4, 2011 г. стр.690-700;
7. Шеховцева, Т.В. определение области эффективного применения станков с ЧПУ в единичном и мелкосерийном производстве деталей ГТД на основе анализатехнологичности их конструкций: дис. канд. техн. наук / Т.В. Шеховцева. – Рыбинск: Рыбинский государственный авиационный технический университет имени П.А. Соловьева, 2012. – 225 с.;

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		63

8. Патент на изобретение RU № 2351441. Способ обработки детали / Кузнецов И.И. Оpubл. 10.04.2009 Бюл. № 10;
9. Патент на изобретение RU № 2355517. Способ и устройство для фрезерования поверхностей произвольной формы / Глэссер Арндт;
10. Патент на изобретение RU № 2356702. Способ и устройство для фрезерования поверхностей произвольной или криволинейной / Глэссер Арндт;
11. Патент на изобретение RU № 2429949. Способ обработки моноколес / Полетаев В.А., Волков Д.И., Клементьев А.В., Плотникова Г.А.;
12. Патент на изобретение RU № 2482940. Способ обработки моноколес / Кобелев С.В., Тукачев Д.В.;
13. Патент на изобретение RU № 2497636. Способ обработки сложных криволинейных поверхностей / Амбросимов С.К., Косенков М.А., Амбросимов К.С., Каптюшина Н.И.;
14. Патент на изобретение RU № 2498883. Способ изготовления лопатки компрессора / Зиновьев Д.В., Пичужкин С.А., Клементьев А.В., Стогов В.С.;
15. Патент на изобретение RU № 2500506. Способ изготовления аэродинамических поверхностей лопаток роторов газотурбинных двигателей на станках с ЧПУ / Барышников О.Е., Вермель В.Д., Болсуновский С.А., Губанов Г.А., Зиняев В.В., Качарава И.Н., Кудряшов А.Б., Шиняев А.В., Шурупов С.В.;
16. Патент на изобретение RU № 2514256. Способ обработки сложных криволинейных поверхностей / Амбросимов С.К., Амбросимов К.С., Бугаков А.И.;
17. Патент на изобретение RU № 2481177. Способ фрезерования на станках с ЧПУ моделей лопаток роторов газотурбинных двигателей /

						Лист
						64
Изм.	Лист	№ докум.	Подпись	Дата		

Барышников О.Е., Вермель В.Д., Мисиюк О.В., Пученков А.Л., Савельев С.В., Шардин А.О., Шиняев А.В.;

18. Яблочников, Е.И. Разработка постпроцессоров для станков с ЧПУ / Яблочников Е.И., Гусельников В.С. – Санкт-Петербург: СПбГУ ИТМО, 2007. – 30 с.;

19. Беднаржевский, В.С. Обзор CAD/CAM/CAE-систем в энергомашиностроении / Беднаржевский В.С. – г. Барнаул: Алтайский государственный технический университет им. И.И. Ползунова;

20. Нискороднов, В. / Мелкосерийное производство корпусных деталей сложной пространственной формы / Нискороднов В. – статья в журнале CADmaster, 2003 №2 – 2 с.;

21. Сазонова, А.О. / Классификация и место САМ-систем в системах автоматизированного проектирования / Сазонова А.О., Дроздов А.А. – г. Пермь: Пермский национальный исследовательский политехнический университет, 2014;

22. Колганов, И. М. / Технологичность авиационных конструкций, пути повышения. Часть1: Учебное пособие / И.М. Колганов, П.В. Дубровский, А. Н. Архипов. – Ульяновск: УлГТУ, 2003. – 148 с.;

23. Демин, Ф.И. / Технология изготовления основных деталей газотурбинных двигателей / Ф.И. Демин, Н.Д. Проничев, И.Л. Шитарев; под. общ. ред. проф. Ф.И. Демина. – Самара: Издательство СГАУ, 2012. – 1 эл. опт. диск (CD-ROM).;

24. Оголь, И.И. / Создание управляющих программ с помощью САМ – систем. / И.И. Оголь. – Томск: Издательство Томского политехнического университета 2014;

25. Ловыгин А. А., Теверовский Л. В. Современный станок с ЧПУ и CAD/CAM-система. – М.: ДМК Пресс, 2012. – 279 с.;

26. Руководство пользователя SprutCAM 8;

27. Руководство пользователя ADEM;

						Лист
						65
Изм.	Лист	№ докум.	Подпись	Дата		

- 28. Руководство пользователя MasterCAM X7;
- 29. Руководство пользователя Delcam PowerMILL.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		66

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

Программа расчета погрешности размера динамической настройки

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApplication3
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            //Ввод исходных данных
            double ebs = Convert.ToDouble(ID1.Text); // интенсивность напряжения
            double Sz = Convert.ToDouble(ID2.Text); // подача на зуб
            double lz = Convert.ToDouble(ID4.Text); // величина износа по задней
поверхности
            double R = Convert.ToDouble(ID5.Text); // радиус фрезы
            double kTr = Convert.ToDouble(ID6.Text); // коэффициент трения
            double Psi = Convert.ToDouble(ID7.Text); // угол контакта
            Psi = Psi * Math.PI / 180;
            double Beta1 = Convert.ToDouble(ID9.Text); // угол сдвига
            Beta1 = Beta1 * Math.PI / 180;
            double Beta = (50 - Beta1) / 0.8;
            double omega = Convert.ToDouble(ID10.Text); // угол наклона
поверхности в направлении движения подачи
            omega = omega * Math.PI / 180;
```

					Лист
Изм.	Лист	№ докум.	Подпись	Дата	67

```

double omegap = Convert.ToDouble(ID11.Text);//угол наклона
поверхности в направлении перпендикулярном направлению движения
подачи
    omegap = omegap * Math.PI / 180;
    double eta = Math.PI / 2 - omegap;
    double B = Convert.ToDouble(ID12.Text); //ширина фрезерования
    double Sper = Convert.ToDouble(ID13.Text);//величина периодической
подачи
    double l = Convert.ToDouble(ID17.Text);// вылет фрезы
    double E = Convert.ToDouble(ID18.Text);// модуль упругости
    double Jpr = Convert.ToDouble(ID19.Text);// момент инерции
    double jzc = Convert.ToDouble(ID20.Text);// жёсткость
    if (radioButton4.Checked == true)
    {
        //Определение нижнего угла контакта
        double alf = Math.PI / 2 - omegap;
        double b1 = 1 / Math.Tan(alf);
        double b2 = Math.Sin(Psi);
        double Fin = Math.Atan(b1 / b2) * 180 / Math.PI;
        Fin = Math.Round(Fin, 3);
        Rs1.Text = "φ(н)=" + Fin.ToString() + "°";
        //Определение верхнего угла профиля
        double a = Math.Cos(Psi) * Math.Sin(alf) * Math.Cos(eta) +
Math.Sin(Psi) * Math.Cos(alf) * Math.Sin(alf);
        double b = Math.Sin(alf) * Math.Sin(eta);
        double c = (1 - B / R) * Math.Sqrt(Math.Pow(Math.Sin(alf), 2) +
Math.Pow(Math.Cos(alf), 2) * Math.Pow(Math.Sin(eta), 2));
        double h1 = (-2 * a + Math.Sqrt(Math.Pow(2 * a, 2) - 4 * (-c - b) * (b -
c))) / (2 * (-c - b));
        double h2 = (-2 * a - Math.Sqrt(Math.Pow(2 * a, 2) - 4 * (-c - b) * (b -
c))) / (2 * (-c - b));
        double Fi1 = Math.Round(2 * Math.Atan(h1) * 180 / Math.PI, 3);
        double Fi2 = Math.Round(2 * Math.Atan(h2) * 180 / Math.PI, 3);
        double Fi = Math.Max(Fi1, Fi2);
        double Fiv = 0;
        // если фрезерование по следу
        if (radioButton1.Checked == true)
        {
            if (Fi > Fin)
            {
                Fiv = Fi;
                Rs2.Text = "φ(в)=" + Fiv.ToString() + "°";
            }
            else

```

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		68


```

    {
        Rs2.Text = "";
        MessageBox.Show("Контакта нет");
    }
}
// если фрезерование по строке
else if (radioButton2.Checked == true)
{
    double[] X = new double[1000];
    Fiv = -Math.PI;
    int i = 0;
    while (Fiv < Math.PI)
    {
        double F = Math.Round(Math.Pow(R * Math.Sin(Fiv) *
Math.Cos(Psi) - Sper * Math.Sin(eta), 2) + Math.Pow(R * Math.Sin(Fiv) *
Math.Sin(Psi) * Math.Cos(alf) + (R * Math.Cos(Fiv) + Sper * Math.Cos(eta)) *
Math.Sin(alf), 2) - R * R, 2);
        if (F == 0)
        {
            X[i] = Fiv;
            i++;
        }
        Fiv = Fiv + 0.0001;
    }
    for (i = 0; i != 1000; i++)
    {
        X[i] = Math.Round(X[i] * 180 / Math.PI, 3);
    }
    for (i = 0; i != 100; i++)
    {
        if (omega >= 0)
        {
            if (X[i] > Fin)
            {
                if (X[i] < 90)
                {
                    if (X[i] < Fi)
                    {
                        Fiv = X[i];
                        Rs2.Text = "φ(в)=" + Fiv.ToString() + "°";
                        if (Fiv > 90)
                        {
                            MessageBox.Show("Режет цилиндрическая часть");
                        }
                    }
                }
            }
        }
    }
}

```

```

        break;
    }
    if (X[i] >= Fi)
    {
        Fiv = Fi;
        Rs2.Text = "φ(в)=" + Fiv.ToString() + "°";
        if (Fiv > 90)
        {
            MessageBox.Show("Режет цилиндрическая часть");
        }
        break;
    }
}
}
}
if (omega < 0)
{
    if (X[i] > Fin)
    {
        if (X[i] <= 90)
        {
            if (X[i] < Fi)
            {
                Fiv = X[i];
                Rs2.Text = "φ(в)=" + Fiv.ToString() + "°";
                break;
            }
            if (X[i] >= Fi)
            {
                Fiv = Fi;
                Rs2.Text = "φ(в)=" + Fiv.ToString() + "°";
                break;
            }
        }
    }
}
}
if (Fiv < Fin)
{
    MessageBox.Show("Контакта нет");
}
}
else

```

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		70

```

{
    MessageBox.Show("Выберите вид фрезерования");
}

//Определение сил резания
double Fz, Fx, Fy, fi, h, sz, sx, sy;
double n = 10000;
Fin = Fin * Math.PI / 180;
Fiv = Fiv * Math.PI / 180;
h = (Fiv - Fin) / n;
//Определение Pz
sz = 0; fi = Fin + h;
while (fi < Fiv)
{
    Fz = 0;
    if (omega >= 0)
    {
        Fz = Math.Sqrt(1 - Math.Pow(Math.Sin(fi) * Math.Sin(Psi) *
Math.Sin(omega) + Math.Cos(fi) * Math.Cos(omega), 2) - Math.Pow(Math.Sin(fi)
* Math.Cos(Psi), 2)) * Math.Cos(Beta) * R / Math.Sin(Beta1);
    }
    else
    {
        Fz = Math.Sqrt(1 - Math.Pow(-Math.Sin(fi) * Math.Sin(Psi) *
Math.Sin(Math.Abs(omega)) + Math.Cos(fi) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(fi) * Math.Cos(Psi), 2)) * Math.Cos(Beta) * R /
Math.Sin(Beta1);
    }
    sz = sz + 4 * Fz;
    fi = fi + h;
    sz = sz + 2 * Fz;
    fi = fi + h;
}
double Fz1 = 0;
double Fz2 = 0;
if (omega >= 0)
{
    Fz1 = Math.Sqrt(1 - Math.Pow(Math.Sin(Fin) * Math.Sin(Psi) *
Math.Sin(omega) + Math.Cos(Fin) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(Fin) * Math.Cos(Psi), 2)) * Math.Cos(Beta) * R /
Math.Sin(Beta1);
    Fz2 = Math.Sqrt(1 - Math.Pow(Math.Sin(Fiv) * Math.Sin(Psi) *
Math.Sin(omega) + Math.Cos(Fiv) * Math.Cos(omega), 2) -

```

```

Math.Pow(Math.Sin(Fiv) * Math.Cos(Psi), 2)) * Math.Cos(Beta) * R /
Math.Sin(Beta1);
    }
    else
    {
        Fz1 = Math.Sqrt(1 - Math.Pow(-Math.Sin(Fin) * Math.Sin(Psi) *
Math.Sin(Math.Abs(omega)) + Math.Cos(Fin) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(Fin) * Math.Cos(Psi), 2)) * Math.Cos(Beta) * R /
Math.Sin(Beta1);
        Fz2 = Math.Sqrt(1 - Math.Pow(-Math.Sin(Fiv) * Math.Sin(Psi) *
Math.Sin(Math.Abs(omega)) + Math.Cos(Fiv) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(Fiv) * Math.Cos(Psi), 2)) * Math.Cos(Beta) * R /
Math.Sin(Beta1);
    }
    sz = h / 3 * (sz + Fz1 - Fz2);
    double Pz = Math.Round((1.15 * ebs * Sz * sz + 0.252 * kTr * ebs * lz *
R * (Fiv - Fin)), 3);
    Rs6.Text = "Pz=" + Pz.ToString() + "H";
    //Определение Px
    sx = 0; fi = Fin + h;
    while (fi < Fiv)
    {
        Fx = 0;
        if (omega >= 0)
        {
            Fx = Math.Sqrt(1 - Math.Pow(Math.Sin(fi) * Math.Sin(Psi) *
Math.Sin(omega) + Math.Cos(fi) * Math.Cos(omega), 2) - Math.Pow(Math.Sin(fi)
* Math.Cos(Psi), 2)) * Math.Cos(Beta) * R / Math.Sin(Beta1);
        }
        else
        {
            Fx = Math.Sqrt(1 - Math.Pow(-Math.Sin(fi) * Math.Sin(Psi) *
Math.Sin(Math.Abs(omega)) + Math.Cos(fi) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(fi) * Math.Cos(Psi), 2)) * Math.Cos(Beta) * R /
Math.Sin(Beta1);
        }
        sx = sx + 4 * Fx;
        fi = fi + h;
        sx = sx + 2 * Fx;
        fi = fi + h;
    }
    double Fx1 = 0;
    double Fx2 = 0;
    if (omega >= 0)

```

```

    {
        Fx1 = Math.Sqrt(1 - Math.Pow(Math.Sin(Fin) * Math.Sin(Psi) *
Math.Sin(omega) + Math.Cos(Fin) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(Fin) * Math.Cos(Psi), 2)) * Math.Sin(Beta) * Math.Cos(Fin)
* R / Math.Sin(Beta1);
        Fx2 = Math.Sqrt(1 - Math.Pow(Math.Sin(Fiv) * Math.Sin(Psi) *
Math.Sin(omega) + Math.Cos(Fiv) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(Fiv) * Math.Cos(Psi), 2)) * Math.Sin(Beta) * Math.Cos(Fiv)
* R / Math.Sin(Beta1);
    }
    else
    {
        Fx1 = Math.Sqrt(1 - Math.Pow(-Math.Sin(Fin) * Math.Sin(Psi) *
Math.Sin(Math.Abs(omega)) + Math.Cos(Fin) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(Fin) * Math.Cos(Psi), 2)) * Math.Sin(Beta) * Math.Cos(Fin)
* R / Math.Sin(Beta1);
        Fx2 = Math.Sqrt(1 - Math.Pow(-Math.Sin(Fiv) * Math.Sin(Psi) *
Math.Sin(Math.Abs(omega)) + Math.Cos(Fiv) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(Fiv) * Math.Cos(Psi), 2)) * Math.Sin(Beta) * Math.Cos(Fiv)
* R / Math.Sin(Beta1);
    }
    sx = h / 3 * (sx + Fx1 - Fx2);
    double Px = Math.Round((1.15 * ebs * Sz * sx + 0.252 * kTr * ebs * lz *
R * (Math.Sin(Fiv) - Math.Sin(Fin))), 3);
    string c4 = Px.ToString();
    Rs4.Text = "Px=" + c4 + "H";
    //Определение Py
    sy = 0; fi = Fin + h;
    while (fi < Fiv)
    {
        Fy = 0;
        if (omega >= 0)
        {
            Fy = Math.Sqrt(1 - Math.Pow(Math.Sin(fi) * Math.Sin(Psi) *
Math.Sin(omega) + Math.Cos(fi) * Math.Cos(omega), 2) - Math.Pow(Math.Sin(fi)
* Math.Cos(Psi), 2)) * Math.Cos(Beta) * R / Math.Sin(Beta1);
        }
        else
        {
            Fy = Math.Sqrt(1 - Math.Pow(-Math.Sin(fi) * Math.Sin(Psi) *
Math.Sin(Math.Abs(omega)) + Math.Cos(fi) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(fi) * Math.Cos(Psi), 2)) * Math.Cos(Beta) * R /
Math.Sin(Beta1);
        }
    }

```

```

        sy = sy + 4 * Fy;
        fi = fi + h;
        sy = sy + 2 * Fy;
        fi = fi + h;
    }
    double Fy1 = 0;
    double Fy2 = 0;
    if (omega >= 0)
    {
        Fy1 = Math.Sqrt(1 - Math.Pow(Math.Sin(Fin) * Math.Sin(Psi) *
Math.Sin(omega) + Math.Cos(Fin) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(Fin) * Math.Cos(Psi), 2)) * Math.Sin(Beta) * Math.Sin(Fin) *
R / Math.Sin(Beta1);
        Fy2 = Math.Sqrt(1 - Math.Pow(Math.Sin(Fiv) * Math.Sin(Psi) *
Math.Sin(omega) + Math.Cos(Fiv) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(Fiv) * Math.Cos(Psi), 2)) * Math.Sin(Beta) * Math.Sin(Fiv) *
R / Math.Sin(Beta1);
    }
    else
    {
        Fy1 = Math.Sqrt(1 - Math.Pow(-Math.Sin(Fin) * Math.Sin(Psi) *
Math.Sin(Math.Abs(omega)) + Math.Cos(Fin) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(Fin) * Math.Cos(Psi), 2)) * Math.Sin(Beta) * Math.Sin(Fin) *
R / Math.Sin(Beta1);
        Fy2 = Math.Sqrt(1 - Math.Pow(-Math.Sin(Fiv) * Math.Sin(Psi) *
Math.Sin(Math.Abs(omega)) + Math.Cos(Fiv) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(Fiv) * Math.Cos(Psi), 2)) * Math.Sin(Beta) * Math.Sin(Fiv) *
R / Math.Sin(Beta1);
    }
    sy = h / 3 * (sx + Fy1 - Fy2);
    double Py = Math.Round((1.15 * ebs * Sz * sy + 0.252 * kTr * ebs * lz *
R * (-Math.Cos(Fiv) + Math.Cos(Fin))), 3);
    Rs5.Text = "Py=" + Py.ToString() + "H";

}
else if (radioButton3.Checked == true)
{
    //Определение нижнего угла контакта
    double teta = Convert.ToDouble(ID16.Text);
    teta = teta * Math.PI / 180;
    double alf = Math.PI / 2 - Math.Atan(Math.Tan(omega) *
Math.Cos(teta));

```

```

double b1 = Math.Cos(teta) / Math.Tan(alf);
double b2 = Math.Sin(Psi);
double Fin = Math.Atan(b1 / b2) * 180 / Math.PI;
Fin = Math.Round(Fin, 3);
Rs1.Text = "φ(н)=" + Fin.ToString() + "°";
//
double lyambda = Math.PI / 2 - alf;
//Определение предельного угла профиля
double dz = Convert.ToDouble(ID14.Text);
double dy = dz * Math.Tan(alf);
double P = Convert.ToDouble(ID12.Text);
double A = Convert.ToDouble(ID15.Text);
A = (A % dy);
//Определение углов профиля горизонтальных участков припуска
double n = 0;
List<double> F1 = new List<double>();
while (n < 10)
{
    if (omega >= 0)
    {
        double Fiv1 = Math.Acos(((R - P) / Math.Sin(alf) + A /
Math.Tan(alf) - dz * n) / R) * 180 / Math.PI;
        if (Fiv1 > Fin)
        {
            if (Fiv1 < 90)
            {
                F1.Add(Fiv1);
            }
        }
        n++;
    }
    else
    {
        double Fiv1 = -Math.Acos(((R - P) / Math.Cos(lyambda) + A *
Math.Tan(Math.Abs(lyambda)) - dz * n) / R) * 180 / Math.PI;
        if (Fiv1 > Fin)
        {
            if (Fiv1 < 90)
            {
                F1.Add(Fiv1);
            }
        }
        n++;
    }
}

```

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		75

```

    }
    F1.Sort();
    for (int i = 0; i < F1.Count; i++)
    {
        textBox1.Text += F1[i].ToString() + Environment.NewLine;
    }
    //определение углов профиля вертикальных участков припуска
    List<double> F2 = new List<double>();
    double n2 = -10;
    while (n2 <= 10)
    {
        if (omega >= 0)
        {
            double Fiv2 = Math.Asin(((dy * n2 / Math.Cos(teta)) - A) / (R *
Math.Sin(Psi + teta))) * 180 / Math.PI;
            if (Fiv2 > Fin)
            {
                if (Fiv2 < 90)
                {
                    F2.Add(Fiv2);
                }
            }
            n2++;
        }
        else
        {
            double Fiv2 = -Math.Asin(((dy * n2 / Math.Cos(teta)) - A) / (R *
Math.Sin(Psi + teta))) * 180 / Math.PI;
            if (Fiv2 > Fin)
            {
                if (Fiv2 < 90)
                {
                    F2.Add(Fiv2);
                }
            }
            n2++;
        }
    }
    F2.Sort();
    for (int i = 0; i < F2.Count; i++)
    {
        textBox2.Text += F2[i].ToString() + Environment.NewLine;
    }
    double Fmax = Math.Max(F1.Max(), F2.Max());

```



```

//определение граничных углов
double n3 = -5;
List<double> F3 = new List<double>();
while (n3 <= 5)
{
    if (omega >= 0)
    {
        double bper = ((n3 - 1) * dy - A) * Math.Sin(Math.PI / 2 - teta) /
Math.Sin(Psi + teta);
        double gper = (R - P) / Math.Sin(alf) + A / Math.Tan(alf) - dz * n3;
        double Figran = Math.Atan(bper / gper) * 180 / Math.PI;
        if (Figran > Fin)
        {
            if (Figran < Fmax)
            {
                F3.Add(Figran);
            }
        }
        n3++;
    }
    else
    {
        double bper = ((n3 - 1) * dy + A) * Math.Sin(Math.PI / 2 - teta) /
Math.Sin(Psi + teta);
        double gper = (R - P) / Math.Sin(alf) + A *
Math.Tan(Math.Abs(Iyambda)) - dz * n3;
        double Figran = Math.Atan(bper / gper) * 180 / Math.PI;
        if (Figran > Fin)
        {
            if (Figran < Fmax)
            {
                F3.Add(Figran);
            }
        }
        n3++;
    }
}
F3.Sort();
for (int i = 0; i < F3.Count; i++)
{
    textBox3.Text += F3[i].ToString() + Environment.NewLine;
}
//Определение граничных углов
double n4 = -5;

```

```

List<double> F4 = new List<double>();
while (n4 <= 5)
{
    if (omega >= 0)
    {
        double bper = (n4 * dy - A) * Math.Sin(Math.PI / 2 - teta) /
Math.Sin(Psi + teta);
        double gper = (R - P) / Math.Sin(alf) + A / Math.Tan(alf) - dz * n4;
        double Figran = Math.Atan(bper / gper) * 180 / Math.PI;
        if (Figran > Fin)
        {
            if (Figran < Fmax)
            {
                F4.Add(Figran);
            }
        }
        n4++;
    }
    else
    {
        double bper = (n4 * dy + A) * Math.Sin(Math.PI / 2 - teta) /
Math.Sin(Psi + teta);
        double gper = (R - P) / Math.Sin(alf) + A *
Math.Tan(Math.Abs(lyambda)) - dz * n4;
        double Figran = Math.Atan(bper / gper) * 180 / Math.PI;
        if (Figran > Fin)
        {
            if (Figran < Fmax)
            {
                F4.Add(Figran);
            }
        }
        n4++;
    }
}
F4.Sort();
for (int i = 0; i < F4.Count; i++)
{
    textBox4.Text += F4[i].ToString() + Environment.NewLine;
}
List<double> FI1 = new List<double>();
List<double> FI2 = new List<double>();
if (P < 0)
{

```

```

if (omega > 0)
{
    if (F2.Min() < F1.Min())
    {
        for (int i = 0; i != 2; i++)
        {
            if (F2[i] < F1[i])
            {
                FI1.Add(F2[i]);
                FI2.Add(F1[i]);
            }
        }
    }
    else
    {
        FI1.Add(Fin);
        FI2.Add(F1.Min());
        for (int i = 0; i != 2; i++)
            if (F2[i] < F1[i + 1])
            {
                FI1.Add(F2[i]);
                FI2.Add(F1[i + 1]);
            }
    }
}
else
{
    if (F1.Min() < F2.Min())
    {
        for (int i = 0; i != 2; i++)
        {
            if (F1[i] < F2[i])
            {
                FI1.Add(F2[i]);
                FI2.Add(F1[i]);
            }
        }
    }
    else
    {
        FI1.Add(Fin);
        FI2.Add(F2.Min());
        for (int i = 0; i != 2; i++)
            if (F2[i] < F1[i])

```

```

        {
            FI1.Add(F1[i]);
            FI2.Add(F2[i]);
        }
    }
}
else
{
    if (omega > 0)
    {
        FI1.Add(Fin);
        FI2.Add(F2.Max());
    }
    else
    {
        if (F3.Min() < F4.Min())
        {
            for (int i = 0; i != 2; i++)
            {
                if (Math.Abs(F2[i]) < Math.Abs(F3[i]))
                {
                    if (Math.Abs(F2[i]) > Math.Abs(F4[i]))
                    {
                        FI2.Add(F2[i]);
                    }
                }
            }
        }
        else
        {
            for (int i = 0; i != 2; i++)
            {
                if (Math.Abs(F2[i]) < Math.Abs(F3[i]))
                {
                    if (Math.Abs(F2[i]) > Math.Abs(F4[i + 1]))
                    {
                        FI2.Add(F2[i]);
                    }
                }
            }
        }
    }
}
}

```

```

for (int i = 0; i < FI1.Count; i++)
{
    textBox5.Text += FI1[i].ToString() + Environment.NewLine;
}
for (int i = 0; i < FI2.Count; i++)
{
    textBox6.Text += FI2[i].ToString() + Environment.NewLine;
}
//Определение сил резания
double Fz, Fx, Fy, fi, h, sz, sx, sy;
double k = 10000;
for (int i = 0; i != FI1.Count; i++)
{
    FI1[i] = FI1[i] * Math.PI / 180;
    FI2[i] = FI2[i] * Math.PI / 180;
}
//Определение Pz
List<double> Pzz = new List<double>();
double Pz = 0;
for (int i = 0; i != FI1.Count; i++)
{
    h = (FI2[i] - FI1[i]) / k;
    sz = 0; fi = FI1[i] + h;
    while (fi < FI2[i])
    {
        Fz = 0;
        if (omega >= 0)
        {
            Fz = Math.Sqrt(1 - Math.Pow(Math.Sin(fi) * Math.Sin(Psi) *
Math.Sin(omega) + Math.Cos(fi) * Math.Cos(omega), 2) - Math.Pow(Math.Sin(fi)
* Math.Cos(Psi), 2)) * Math.Cos(Beta) * R / Math.Sin(Beta1);
        }
        else
        {
            Fz = Math.Sqrt(1 - Math.Pow(-Math.Sin(fi) * Math.Sin(Psi) *
Math.Sin(Math.Abs(omega)) + Math.Cos(fi) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(fi) * Math.Cos(Psi), 2)) * Math.Cos(Beta) * R /
Math.Sin(Beta1);
        }
        sz = sz + 4 * Fz;
        fi = fi + h;
        sz = sz + 2 * Fz;
        fi = fi + h;
    }
}

```

```

double Fz1 = 0;
double Fz2 = 0;
if (omega >= 0)
{
    Fz1 = Math.Sqrt(1 - Math.Pow(Math.Sin(FI1[i]) * Math.Sin(Psi) *
Math.Sin(omega) + Math.Cos(FI1[i]) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(FI1[i]) * Math.Cos(Psi), 2)) * Math.Cos(Beta) * R /
Math.Sin(Beta1);
    Fz2 = Math.Sqrt(1 - Math.Pow(Math.Sin(FI2[i]) * Math.Sin(Psi) *
Math.Sin(omega) + Math.Cos(FI2[i]) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(FI2[i]) * Math.Cos(Psi), 2)) * Math.Cos(Beta) * R /
Math.Sin(Beta1);
}
else
{
    Fz1 = Math.Sqrt(1 - Math.Pow(-Math.Sin(FI1[i]) * Math.Sin(Psi) *
Math.Sin(Math.Abs(omega)) + Math.Cos(FI1[i]) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(FI1[i]) * Math.Cos(Psi), 2)) * Math.Cos(Beta) * R /
Math.Sin(Beta1);
    Fz2 = Math.Sqrt(1 - Math.Pow(-Math.Sin(FI2[i]) * Math.Sin(Psi) *
Math.Sin(Math.Abs(omega)) + Math.Cos(FI2[i]) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(FI2[i]) * Math.Cos(Psi), 2)) * Math.Cos(Beta) * R /
Math.Sin(Beta1);
}
sz = h / 3 * (sz + Fz1 - Fz2);
Pzz.Add(Math.Round((1.15 * ebs * Sz * sz + 0.252 * kTr * ebs * lz *
R * (FI2[i] - FI1[i])), 3));
Pz += Pzz[i];
}
Rs6.Text = "Pz=" + Pz.ToString() + "H";
//Определение Px
List<double> Pxx = new List<double>();
double Px = 0;
for (int i = 0; i != FI1.Count; i++)
{
    h = (FI2[i] - FI1[i]) / k;
    sx = 0; fi = FI1[i] + h;
    while (fi < FI2[i])
    {
        Fx = 0;
        if (omega >= 0)
        {

```

```

        Fx = Math.Sqrt(1 - Math.Pow(Math.Sin(fi) * Math.Sin(Psi) *
Math.Sin(omega) + Math.Cos(fi) * Math.Cos(omega), 2) - Math.Pow(Math.Sin(fi)
* Math.Cos(Psi), 2)) * Math.Cos(Beta) * R / Math.Sin(Beta1);
    }
    else
    {
        Fx = Math.Sqrt(1 - Math.Pow(-Math.Sin(fi) * Math.Sin(Psi) *
Math.Sin(Math.Abs(omega)) + Math.Cos(fi) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(fi) * Math.Cos(Psi), 2)) * Math.Cos(Beta) * R /
Math.Sin(Beta1);
    }
    sx = sx + 4 * Fx;
    fi = fi + h;
    sx = sx + 2 * Fx;
    fi = fi + h;
}
double Fx1 = 0;
double Fx2 = 0;
if (omega >= 0)
{
    Fx1 = Math.Sqrt(1 - Math.Pow(Math.Sin(FI1[i]) * Math.Sin(Psi) *
Math.Sin(omega) + Math.Cos(FI1[i]) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(FI1[i]) * Math.Cos(Psi), 2)) * Math.Sin(Beta) *
Math.Cos(FI1[i]) * R / Math.Sin(Beta1);
    Fx2 = Math.Sqrt(1 - Math.Pow(Math.Sin(FI2[i]) * Math.Sin(Psi) *
Math.Sin(omega) + Math.Cos(FI2[i]) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(FI2[i]) * Math.Cos(Psi), 2)) * Math.Sin(Beta) *
Math.Cos(FI2[i]) * R / Math.Sin(Beta1);
}
else
{
    Fx1 = Math.Sqrt(1 - Math.Pow(-Math.Sin(FI1[i]) * Math.Sin(Psi) *
Math.Sin(Math.Abs(omega)) + Math.Cos(FI1[i]) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(FI1[i]) * Math.Cos(Psi), 2)) * Math.Sin(Beta) *
Math.Cos(FI1[i]) * R / Math.Sin(Beta1);
    Fx2 = Math.Sqrt(1 - Math.Pow(-Math.Sin(FI2[i]) * Math.Sin(Psi) *
Math.Sin(Math.Abs(omega)) + Math.Cos(FI2[i]) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(FI2[i]) * Math.Cos(Psi), 2)) * Math.Sin(Beta) *
Math.Cos(FI2[i]) * R / Math.Sin(Beta1);
}
    sx = h / 3 * (sx + Fx1 - Fx2);
    Pxx.Add(Math.Round((1.15 * ebs * Sz * sx + 0.252 * kTr * ebs * lz *
R * (Math.Sin(FI2[i]) - Math.Sin(FI1[i]))), 3));
    string c4 = Px.ToString();

```

```

    Px += Pxx[i];
}
Rs4.Text = "Px=" + Px.ToString() + "H";
//Определение Py
List<double> Pyy = new List<double>();
double Py = 0;
for (int i = 0; i != FI1.Count; i++)
{
    h = (FI2[i] - FI1[i]) / k;
    sy = 0; fi = Fin + h;
    while (fi < FI2[i])
    {
        Fy = 0;
        if (omega >= 0)
        {
            Fy = Math.Sqrt(1 - Math.Pow(Math.Sin(fi) * Math.Sin(Psi) *
Math.Sin(omega) + Math.Cos(fi) * Math.Cos(omega), 2) - Math.Pow(Math.Sin(fi)
* Math.Cos(Psi), 2)) * Math.Cos(Beta) * R / Math.Sin(Beta1);
        }
        else
        {
            Fy = Math.Sqrt(1 - Math.Pow(-Math.Sin(fi) * Math.Sin(Psi) *
Math.Sin(Math.Abs(omega)) + Math.Cos(fi) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(fi) * Math.Cos(Psi), 2)) * Math.Cos(Beta) * R /
Math.Sin(Beta1);
        }
        sy = sy + 4 * Fy;
        fi = fi + h;
        sy = sy + 2 * Fy;
        fi = fi + h;
    }
    double Fy1 = 0;
    double Fy2 = 0;
    if (omega >= 0)
    {
        Fy1 = Math.Sqrt(1 - Math.Pow(Math.Sin(FI1[i]) * Math.Sin(Psi) *
Math.Sin(omega) + Math.Cos(FI1[i]) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(FI1[i]) * Math.Cos(Psi), 2)) * Math.Sin(Beta) *
Math.Sin(FI1[i]) * R / Math.Sin(Beta1);
        Fy2 = Math.Sqrt(1 - Math.Pow(Math.Sin(FI2[i]) * Math.Sin(Psi) *
Math.Sin(omega) + Math.Cos(FI2[i]) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(FI2[i]) * Math.Cos(Psi), 2)) * Math.Sin(Beta) *
Math.Sin(FI2[i]) * R / Math.Sin(Beta1);
    }
}

```



```

else
{
    Fy1 = Math.Sqrt(1 - Math.Pow(-Math.Sin(FI1[i]) * Math.Sin(Psi) *
Math.Sin(Math.Abs(omega)) + Math.Cos(FI1[i]) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(FI1[i]) * Math.Cos(Psi), 2)) * Math.Sin(Beta) *
Math.Sin(FI1[i]) * R / Math.Sin(Beta1);
    Fy2 = Math.Sqrt(1 - Math.Pow(-Math.Sin(FI2[i]) * Math.Sin(Psi) *
Math.Sin(Math.Abs(omega)) + Math.Cos(FI2[i]) * Math.Cos(omega), 2) -
Math.Pow(Math.Sin(FI2[i]) * Math.Cos(Psi), 2)) * Math.Sin(Beta) *
Math.Sin(FI2[i]) * R / Math.Sin(Beta1);
}
sy = h / 3 * (sy + Fy1 - Fy2);
Pyy.Add(Math.Round((1.15 * ebs * Sz * sy + 0.252 * kTr * ebs * lz *
R * (-Math.Cos(FI2[i]) + Math.Cos(FI1[i]))), 3));
Py += Pyy[i];
}
Rs5.Text = "Py=" + Py.ToString() + "H";

double Px_pop = Pz * Math.Sin(Psi) - Py * Math.Cos(Psi);
double Py_pr = Pz * Math.Cos(Psi) + Py * Math.Sin(Psi);
double Pz_oz = Px;

double deltaN = ((Px_pop * Math.Tan(omega) + Py_pr *
Math.Tan(omegap)) * 1 * 1 * 1 / (3 * E * Jpr) - Pz_oz / jzc) /
(Math.Sqrt(Math.Tan(omega) * Math.Tan(omega) + Math.Tan(omegap) *
Math.Tan(omegap) + 1));

Rs7.Text = "Px_поп = " + Px_pop + "H";
Rs8.Text = "Py_пр = " + Py_pr + "H";
Rs9.Text = "Pz_ос = " + Pz_oz + "H";
Rs10.Text = "ΔN = " + deltaN + "мм";
}
else
{
    MessageBox.Show("Выберите вид припуска");
}
}
private void button2_Click(object sender, EventArgs e)
{
    ID1.Text = "0";
    ID2.Text = "0";
    ID4.Text = "0";
    ID5.Text = "8";
}

```

```

ID6.Text = "0";
ID7.Text = "90";
ID9.Text = "30";
ID10.Text = "30";
ID11.Text = "0";
ID12.Text = "1";
ID13.Text = "4";
Rs1.Text = "";
Rs2.Text = "";
Rs5.Text = "";
Rs4.Text = "";
Rs6.Text = "";
textBox1.Text = "";
textBox2.Text = "";
textBox3.Text = "";
textBox4.Text = "";
textBox5.Text = "";
textBox6.Text = "";
}
private void radioButton1_CheckedChanged(object sender, EventArgs e)
{
    ID13.Visible = false;
    Label51.Text = "";
    label5.Text = "";
}
private void radioButton2_CheckedChanged(object sender, EventArgs e)
{
    ID13.Visible = true;
    Label51.Text = "Спер=";
    label5.Text = "-величина периодической подачи";
}

private void radioButton3_CheckedChanged(object sender, EventArgs e)
{
    Label50.Text = "P=";
    label4.Text = "- величина сплошного припуска без учета
ступенчатого";
    label10.Text = "- размер ступени";
    label11.Text = "dz=";
    label14.Text = "- смещение инструмента";
    label15.Text = "A=";
    label16.Text = "- направление подачи";
    label17.Text = "θ=";
    ID16.Visible = true;
}

```

```

        ID14.Visible = true;
        ID15.Visible = true;
    }

private void radioButton4_CheckedChanged(object sender, EventArgs e)
{
    label10.Text = "";
    label11.Text = "";
    label14.Text = "";
    label15.Text = "";
    label16.Text = "";
    label17.Text = "";
    ID16.Visible = false;
    ID14.Visible = false;
    ID15.Visible = false;
    Label50.Text = "В=";
    label4.Text = "- ширина фрезерования";
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
}
}
}
}

```

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		87

ПРИЛОЖЕНИЕ Б
Публикация по теме ВКР

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		88

ПРИЛОЖЕНИЕ В

Протокол проверки на антиплагиат

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		89