

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Белорусский национальный технический университет

Кафедра «Программное обеспечение информационных систем
и технологий»

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
В АВТОМАТИЗИРОВАННОМ ПРОЕКТИРОВАНИИ
МАШИНОСТРОИТЕЛЬНЫХ КОНСТРУКЦИЙ

Учебно-методическое пособие
для студентов специальности 1-40 05 01-01
«Информационные системы технологии
(в проектировании и производстве)»

В 3 частях

Часть 1

*Рекомендовано учебно-методическим объединением по образованию
в области информатики и радиоэлектроники*

Минск
БНТУ
2020

УДК 621-025.13:044.9(075.8)

ББК 34.4-05я7

И74

Авторы:

*А. В. Бородуля, И. Л. Ковалева, В. А. Кочуров, С. В. Красновская,
Д. П. Кункевич, В. В. Напрасников, Ю. В. Полозков*

Рецензенты:

зав. кафедрой, канд. техн. наук, доцент *М. С. Шмаков*,
канд. техн. наук, доцент *В. С. Юденков*;
зав. лабораторией моделирования технологических процессов
ОИПИ НАН Беларуси *А. Г. Гривочевский*

Информационные технологии в автоматизированном проектировании машиностроительных конструкций : учебно-методическое пособие для студентов специальности 1-40 05 01-01 «Информационные системы и технологии (в проектировании и производстве)» : в 3 ч. / А. В. Бородуля [и др.]. – Минск : БНТУ, 2020. – Ч. 1. – 98 с.
ISBN 978-985-583-166-3 (Ч. 1).

Настоящий материал предназначен для использования в качестве методических указаний при выполнении лабораторных, курсовых и дипломных работ студентами направления специальности 1-40 05 01-01 «Информационные системы и технологии (в проектировании и производстве)» по дисциплинам «Автоматизация технологического проектирования», «Компьютерные системы конечно-элементных расчетов», «Оптимизация проектных решений».

Указания будут полезны для магистрантов и аспирантов технических специальностей, научные интересы которых связаны с конечно-элементным моделированием.

УДК 621-025.13:044.9(075.8)

ББК 34.4-05я7

ISBN 978-985-583-166-3 (Ч. 1)
ISBN 978-985-583-167-0

© Белорусский национальный
технический университет, 2020

ОГЛАВЛЕНИЕ

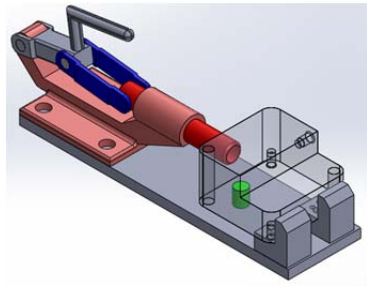
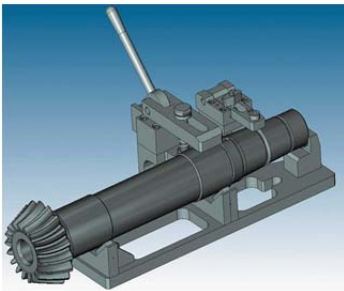
РАЗДЕЛ 1. АВТОМАТИЗАЦИЯ КОНСТРУКТОРСКОГО ПРОЕКТИРОВАНИЯ.....	6
Введение.....	7
1. Проектирование приспособления для установки и закрепления деталей.....	8
1.1. Цели и задачи.....	8
1.2. Объект проектирования.....	8
1.3. Методика проектирования приспособления.....	11
1.4. Размещение функциональных элементов.....	12
1.5. Проектирование корпуса.....	14
1.6. Задания и средства выполнения.....	16
2. Интерфейс программирования приложений пакета геометрического моделирования.....	17
2.1. Начало работы.....	18
2.2. Макрокоманды.....	18
2.3. Анализ и редактирование макроса.....	18
2.4. Справка по API.....	19
2.5. Процедуры открытия и закрытия файла компонента.....	20
2.6. Программирование на уровне детали.....	20
3. Автоматизация сопряжений.....	22
3.1. Цель и задачи.....	22
3.2. Создание макрокоманды.....	23
3.3. Анализ полученного макроса.....	24
3.4. Процедура сопряжения.....	25
4. Извлечение информации из геометрических моделей в твердотельном формате.....	26
4.1. Цель работы и решаемые задачи.....	26
4.2. Твердотельная модель.....	27
4.3. Доступ к элементам твердотельной модели.....	29
4.4. Использование геометрической информации в процедурах.....	31
РАЗДЕЛ 2. КОНЕЧНО-ЭЛЕМЕНТНОЕ МОДЕЛИРОВАНИЕ СЛОЖНЫХ СТЕРЖНЕВЫХ КОНСТРУКЦИЙ В ANSYSWORKBENCH.....	33
Введение.....	34
1. Описание конструкции.....	34

2. Постановка задачи	35
3. Построение геометрической модели башни крана.....	37
3.1. Создание блока геометрии на схеме проекта	38
3.2. Определение сечений стержней.....	40
3.3. Создание параметрической геометрической модели.....	42
3.3.1. Построение отдельной секции.....	42
3.3.2. Построение секционной части башни.....	53
3.3.3. Построение площадки под навесную кабину.....	53
3.3.4. Построение распора башни.....	55
3.3.5. Построение оголовка башни	56
3.3.6. Построение тросов башни.....	57
3.3.7. Заключительный этап создания геометрии башни. Объединение всех тел в единую конструкцию	57
4. Построение конечно-элементной модели башни крана и проведение статического анализа конструкции	58
4.1. Подготовка конечно-элементной модели башни	58
4.1.1. Создание блока анализа на схеме проекта	58
4.1.2. Учет навесной кабины на башне крана с помощью «точечной массы»	59
4.1.3. Учет стрелы крана с помощью «удаленной точки».....	60
4.1.4. Учет ветровой нагрузки на кран.....	61
4.1.5. Учет тросов крана	61
4.2. Задание граничных условий.....	63
4.2.1. Закрепление опор башни.....	63
4.2.2. Собственный вес конструкции	63
4.2.3. Противовес	64
4.2.4. Ветровая нагрузка.....	64
4.2.5. Нагрузка от поднимаемого груза.....	66
4.3. Проведение статического анализа конструкции	66
4.4. Анализ потери устойчивости	67
4.5. Модальный анализ	69
4.6. Гармонический анализ.....	71
ПРИЛОЖЕНИЕ 1. Схема проекта	74
ПРИЛОЖЕНИЕ 2. Формы собственных колебаний	75
ПРИЛОЖЕНИЕ 3. Задания по вариантам.....	76
РАЗДЕЛ 3. ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ	77
Введение.....	78

1. Генетические алгоритмы	78
2. Классический ГА и его модификации	79
2.1. Классический ГА	79
2.2. Варианты реализации основных этапов классического ГА.....	81
2.2.1. Инициализация, или формирование исходной популяции	81
2.2.2. Селекция хромосом (отбор)	82
2.2.3. Операторы кроссинговера	84
2.2.4. Операторы мутации	86
2.2.5. Стратегии формирования нового поколения	87
2.2. ГА с изменяемой мощностью популяции.....	88
2.3. Последовательные модификации классического ГА	89
2.3.1. Алгоритм Genitor	89
2.3.2. Алгоритм SteadyState (SSGA).....	89
2.3.3. ГА селекционеров.....	89
2.3.4. Алгоритм СНС	89
2.3.5. Гибридный ГА	90
2.4. Параллельные модификации классического ГА.....	90
2.4.1. Параллельный ГА на базе популяций.....	91
2.4.2. Миграция	91
2.4.3. Модель «Рабочий – Хозяин»	92
2.4.4. Диффузионная модель	92
3. Основные этапы и требования к выполняемой работе	92
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	97

Раздел 1

АВТОМАТИЗАЦИЯ КОНСТРУКТОРСКОГО ПРОЕКТИРОВАНИЯ



ВВЕДЕНИЕ

В данном разделе рассматриваются вопросы, связанные с разработкой конструкторских приложений на базе универсальной системы геометрического моделирования. Конструкторское проектирование – процесс творческий, трудно формализуемый. Поэтому основные средства его автоматизации – системы геометрического моделирования – КОМПАС 3D, SolidWorks, NX и т. п. Они позволяют быстро визуализировать решение в виде объемной модели, подготовить конструкторскую документацию, выполнить инженерный анализ и др. Наряду с творческими задачами конструктору приходится выполнять много рутинных процедур, связанных с описанием типовых решений. Соответствующие программные средства были бы важным подспорьем в его работе. В пользу подобных средств говорит и имеющая место тенденция к разработке проблемно ориентированных модулей на базе универсальных систем: «Mold Wizard» (проектирование пресс-форм) и «Progressive Die Wizard» (штампы последовательного действия) в системе NX, «РЕДУКТОР-3D V2.3» (одноступенчатые редукторы) в системе КОМПАС 3D и др.

В качестве объекта проектирования предлагается приспособление для установки и закрепления деталей при выполнении над ними различных технологических операций. В первой работе на данном примере предлагается ознакомиться с основными принципами автоматизированного конструирования. Вторая работа посвящена ознакомлению с программным интерфейсом системы геометрического моделирования на примере вставки конструктивного элемента в сборочную модель. В третьей работе рассматриваются способы решения задач позиционирования при помощи сопряжений. Четвертая работа – ознакомление с внутренней структурой твердотельной геометрической модели и способами извлечения информации из нее.

1. ПРОЕКТИРОВАНИЕ ПРИСПОСОБЛЕНИЯ ДЛЯ УСТАНОВКИ И ЗАКРЕПЛЕНИЯ ДЕТАЛЕЙ

1.1. Цели и задачи

В «конструирующие» программы¹ при разработке закладываются определенные технологии конструирования, в основе которых лежат традиционные методики проектирования, используемые конструкторами. С другой стороны, неотъемлемой составляющей современного конструирования являются **системы геометрического моделирования** – «SolidWorks», КОМПАС 3D и т. п. Сочетание возможностей этих систем с традиционными методиками конструирования обеспечивает эффективную программную поддержку конструирования соответствующего класса объектов. Отработать логику программ позволяет интерактивное проектирование в геометрической системе, соблюдая соответствующие методики способствует.

Цель работы: ознакомиться с основными принципами конструирования технических объектов с использованием средств автоматизации проектирования. Для достижения поставленной цели предлагается решить задачу проектирования объекта заданного класса, используя соответствующие методические указания и систему геометрического моделирования.

1.2. Объект проектирования

В качестве объекта проектирования предлагаются приспособления для установки и закрепления деталей при выполнении над ними различных технологических операций – механической обработки, сборки и т. п.

Деталь, которая устанавливается и закрепляется, называется **объектом оснащения**. Пусть в качестве такого объекта задана деталь **Д** (рис. 1.1). В ней необходимо прорезать уступ шириной **В** и глубиной **h**. Уступ можно вырезать инструментом **И** (рис. 1.2), инструмент надо закрепить в станке (рис. 1.3, *а*). Деталь надо также закрепить на станке. Причем сделать это надо так, чтобы пересечение инструмента и детали имело размеры паза. Для этого на столе станка крепится

¹ Программы, поддерживающие генерирование конструкторских решений.

опора **О** так, чтобы ее верхняя грань находилась на расстоянии $H-h$ от нижнего торца инструмента (рис. 1.3, б). Там же на столе устанавливается упор **У** на расстоянии **В** от внешнего края инструмента. Любая деталь, аналогичная детали **Д**, устанавливается поверхностью **Б1** на опору **О** и упирается поверхностью **Б2** в упор **У**. Инструмент **И** при этом вырежет в детали именно такой паз, который нужен.

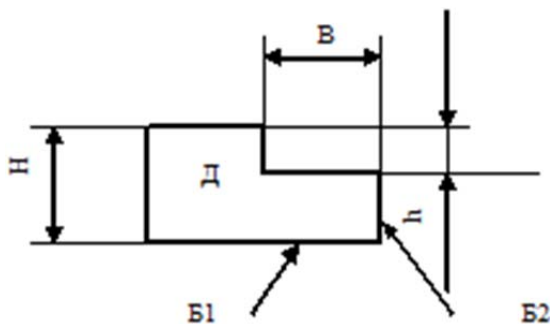


Рис. 1.1. Деталь, размерная схема

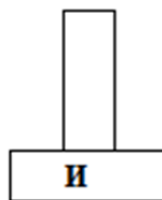


Рис. 1.2. Инструмент

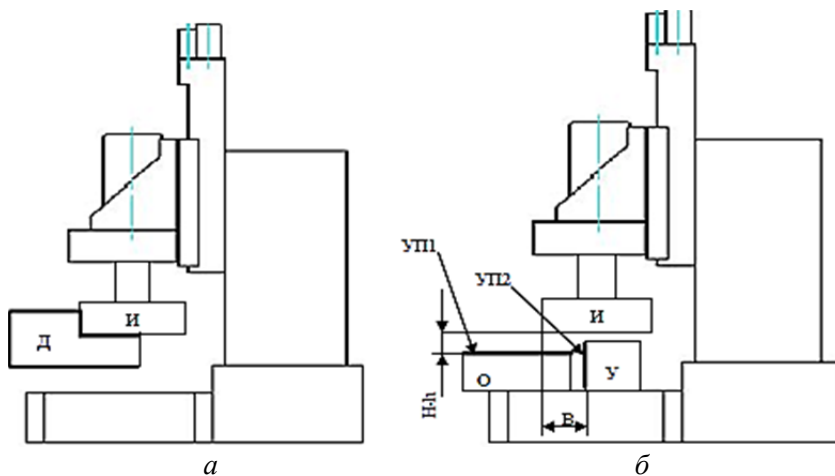


Рис. 1.3. Размещение детали относительно инструмента (а) и приспособление (б)

В приведенном примере приспособление – это опора **О** и упор **У**. В действительности опора **О** и упор **У** – только части приспособления, так называемые установочные элементы. Кроме них необходимы, например, закрепляющие элементы.

Установочные и закрепляющие элементы непосредственно реализуют функцию приспособления, поэтому они называются функциональными.

Функциональные элементы объединяются корпусом. На рисунке он не показан, но он обязательно должен быть. На нем собираются все детали приспособления, и оно устанавливается на оборудование именно при помощи корпуса.

Таким образом, структура приспособления включает элементы установочные, закрепляющие и корпус (рис. 1.4). Это минимальная структура. Кроме того, в приспособлении могут быть различные вспомогательные элементы.

Поверхности детали, которыми она устанавливается в приспособление, называются «базы». В приведенном примере (рис. 1.1) базами являются поверхность **Б1**, которой деталь ложится на опору, и поверхность **Б2**, которой деталь упирается в упор.

Поверхности установочных элементов (см. рис. 1.1), на которые устанавливается (**УП1**) деталь и в которые она упирается (**УП2**) называются «установочные поверхности приспособления».

На рис. 1.4 приведен пример приспособления рассмотренного случая.

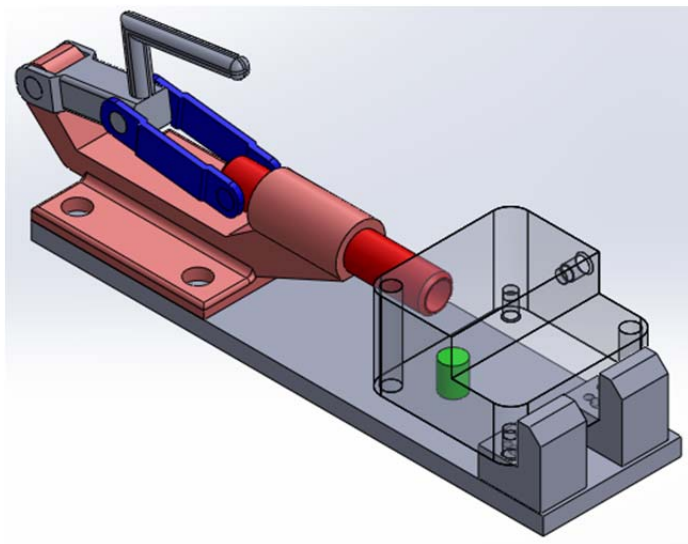


Рис. 1.4. Внешний вид приспособления для прорезания паза

1.3. Методика проектирования приспособления

Проектирование приспособлений, в общем случае, задача сложная и творческая. Они достаточно разнообразны, что обусловлено разновидностью деталей – объектов оснащения. Тем не менее, структура этих объектов подчиняется определенным закономерностям. Эти два фактора: разнообразие и наличие закономерностей – делают приспособления интересным объектом автоматизации конструкторского проектирования.

Исходная информация для проектирования приспособления – описание детали – объекта оснащения. Документальное его описание традиционно представляется в виде конструкторского чертежа. В данной работе для проектирования используется система геометрического моделирования. Соответственно, описание детали – геометрическая модель в **твердотельном формате**.

Приспособление – сборный объект. Соответственно, необходимо создать новый документ, поддерживающий формирование сборочных единиц. В этот документ в качестве первого компонента вставляется модель объекта оснащения (рис. 1.5, *а*).

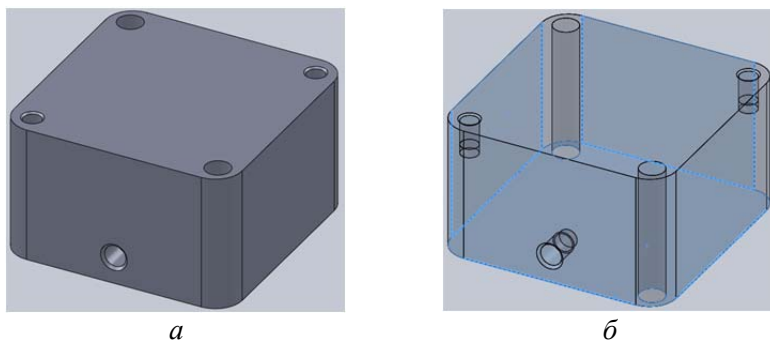


Рис. 1.5. Внешний вид (*а*) и базовые поверхности (*б*) объекта оснащения

Основная характеристика объекта оснащения – комплект баз. Для установки любой детали, в общем случае, необходимо и достаточно **три базы**. В большинстве случаев базы – это плоские или цилиндрические поверхности. Выбор баз определяется технологическим процессом в целом. Поэтому в рамках данной работы базы считаются заданными (рис. 1.5, *б*).

1.4. Размещение функциональных элементов

Функциональными считаются установочные и закрепляющие элементы.

Установочные элементы размещаются относительно баз. Эта процедура включает две задачи: выбор элементов и их позиционирование.

Принцип выбора элементов в рамках данной работы простой. Среди установочных элементов можно выделить несколько наиболее распространенных классов:

- **опора** – элемент с **горизонтальной** установочной поверхностью (рис. 1.6, *а*, *б*);
- **упор** – элемент с **вертикальной** установочной поверхностью, в качестве которого можно использовать цилиндрический палец (рис. 1.7, *а*);
- **палец** – элемент с **цилиндрической** установочной поверхностью (рис. 1.7).

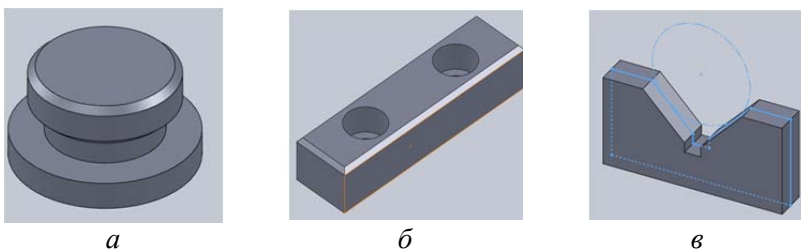


Рис. 1.6. Опоры установочные:
а – с плоской головкой; *б* – пластина опорная; *в* – опорная призма

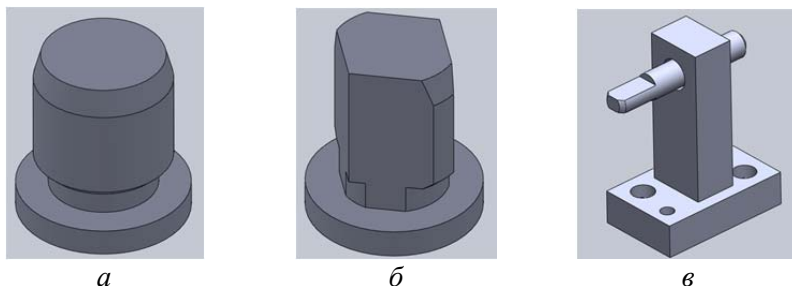


Рис. 1.7. Пальцы установочные:
а – цилиндрический; *б* – срезанный; *в* – выдвижной

Для горизонтальных баз выбираются опоры, для вертикальных – упоры, для отверстий – пальцы. Если горизонтальная база – цилиндр, следует использовать опору призматическую. Если имеется два базовых отверстия, один из пальцев должен быть срезанным. Если оси отверстий не параллельны, один из пальцев (сбоку) должен быть выдвижным.

При расстановке установочных элементов в рамках данной работы также можно руководствоваться простейшим принципом устойчивости.

Для этой устойчивости, прежде всего, необходимо обеспечить горизонтальную установочную поверхность. Она должна быть достаточно развитой, чтобы деталь не качалась. **Сплошная** поверхность **не годится** – это не технологично. Следовательно, она должна быть фрагментальной. Как известно из геометрии, плоскость определяется тремя точками. Поэтому для плоской горизонтальной базы следует выбрать три опоры и расставить их пошире (рис. 1.8, *а*).

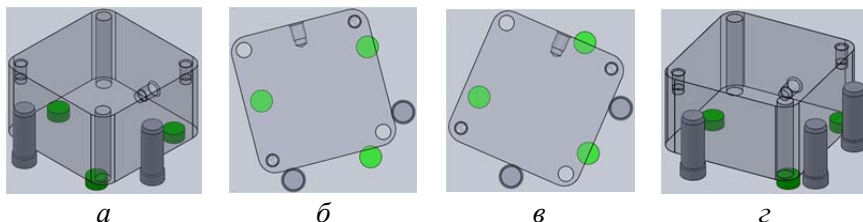


Рис. 1.8. Схема установки:
а – опоры и два упора; *б, в* – отклонения от заданного положения;
г – окончательный вариант

Далее необходимо ограничить перемещения вдоль горизонтальной установочной поверхности. Для этого против каждой вертикальной базы устанавливается упор (см. рис. 1.8, *а*). Однако относительно этих упоров деталь может перемещаться (рис. 1.8, *б, в*).

Чтобы добиться однозначности положения необходимо для одной базы добавить второй упор (рис. 1.8, *г*).

Если два установочных элемента расположены достаточно близко, можно использовать один **комбинированный** элемент, например упор уголкового (рис. 1.9, *а*). Таким элементом можно заменить пару «опора–упор» с правой стороны (рис. 1.9, *б*). А вот замена оставшейся пары комбинированным упором (рис. 1.9, *в*) не допу-

стима – опорная точка слишком смещается от своего положения, теряется устойчивость.

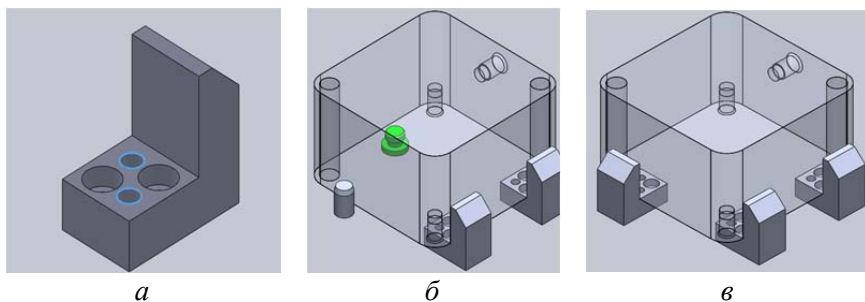


Рис. 1.9. Комбинированный элемент:
a – внешний вид; *б* – направляющая база; *в* – опорная и направляющая базы

Фиксирующие элементы располагаются так, чтобы их усилие было направлено на установочные элементы (рис. 1.10).

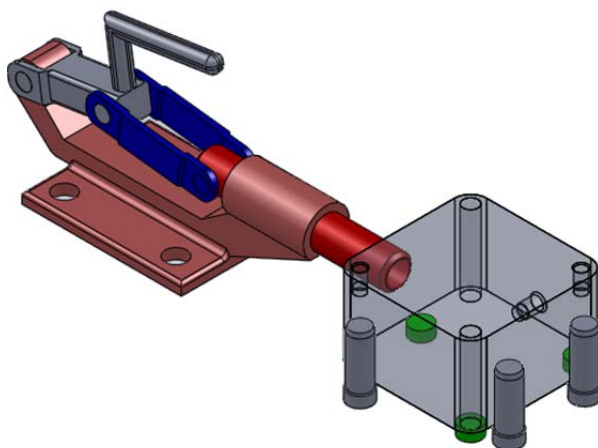


Рис. 1.10. Размещение фиксирующего элемента

1.5. Проектирование корпуса

Корпус приспособления проектируется после расстановки функциональных элементов. Ограничений на конструкцию этого элемента практически нет. Поэтому он может быть любым, главное, чтобы соединял между собой функциональные элементы.

В данном контексте необходимо ввести понятия **привалочная грань** и **привалочная плоскость**. В конструкции каждого элемента (детали) есть грань, которой он опирается на корпус (рис. 1.11). Это и есть привалочная грань конструктивного элемента.

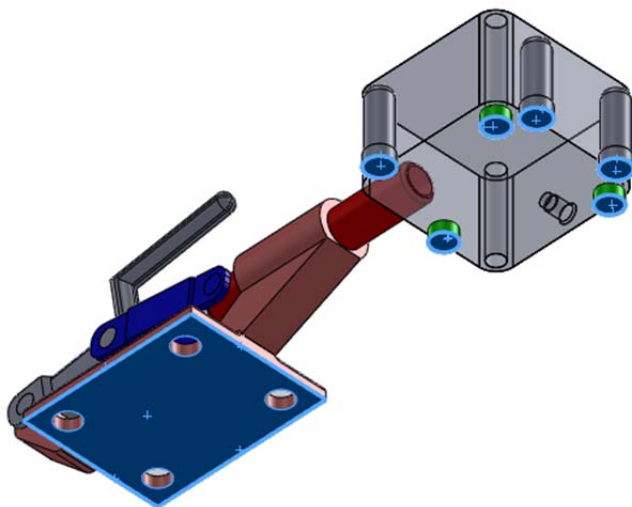
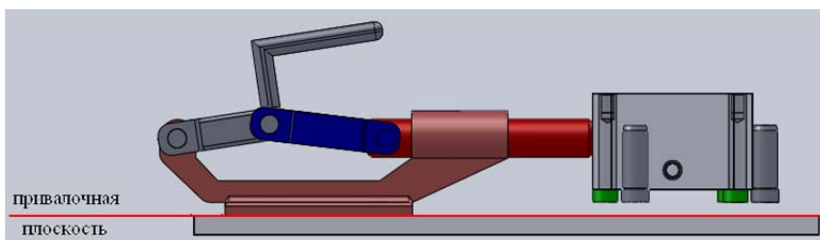


Рис. 1.11. Привалочные грани функциональных элементов

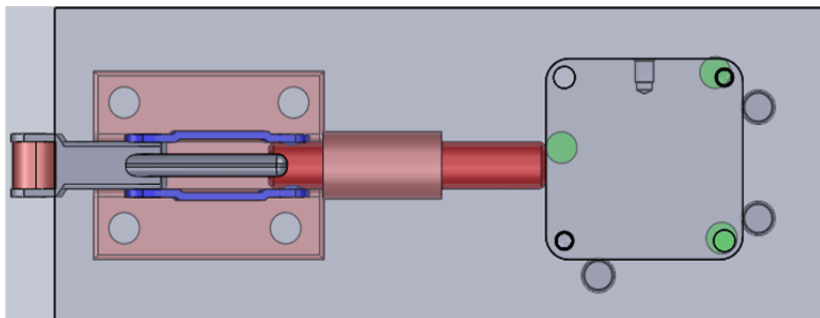
Элементов в конструкции много. Соответственно, много и привалочных граней. Плоскость, **параллельная** привалочным граням и расположенная **ниже всех** граней, будет называться **привалочной плоскостью**. В приведенном примере ниже всех располагается привалочная грань зажима (рис. 1.12, *a*). С ней и будет связана привалочная плоскость.

Простейший вариант корпуса – прямоугольная плита. Верхняя грань плиты совпадает с привалочной плоскостью (см. рис. 1.12, *a*). Габариты и расположение плиты «в плане» таковы, чтобы ее внешний контур охватывал проекции всех привалочных граней на привалочную плоскость (рис. 1.12, *b*).

Промежутки между привалочными гранями функциональных элементов и корпусной плитой должны быть заполнены. Для этого надо либо удлинить элемент, либо вставить между ним и корпусной плитой переходник.



а



б

Рис. 1.12. Корпусная плата

1.6. Задания и средства выполнения

Проектирование приспособления выполняется при помощи средств геометрического моделирования. Можно выбрать любую систему, поддерживающую **твердотельное моделирование**. Рекомендуемый пакет **SolidWorks**. В последующих работах планируется разработка **программных средств** поддержки проектирования. Соответственно, необходима графическая система, предоставляющая эффективный интерфейс разработки пользовательских приложений. **SolidWorks** – один из лучших представителей таких систем.

Ознакомиться с набором команд, минимально необходимым для компоновки модели приспособления, поможет учебное пособие: *«Учебные пособия SolidWorks» > «Введение» > «Упражнение 2 – Сборки»*.

Для создания моделей отдельных деталей (например, при разработке собственных элементов) полезным будет пособие: *«Учебные пособия SolidWorks» > «Введение» > «Упражнение 1 – Детали»*.

Исходные данные для выполнения лабораторной работы – геометрическая модель детали в формате **STEP** (**ST**andard for **E**xchange of **P**roduct model data) с выделенными базовыми поверхностями (рис. 1.13). Перед началом работы каждую такую модель необходимо импортировать в SolidWorks и сохранить.



Рис. 1.13. Геометрическая модель детали с выделенными базами

Элементы приспособления также предоставляются в готовом виде. Однако можно создавать и собственные. Все элементы, используемые при выполнении работы, должны быть скопированы в собственный раздел. Не следует использовать ссылки на модели из указанного «общего» каталога.

2. ИНТЕРФЕЙС ПРОГРАММИРОВАНИЯ ПРИЛОЖЕНИЙ ПАКЕТА ГЕОМЕТРИЧЕСКОГО МОДЕЛИРОВАНИЯ

Цель работы: ознакомиться с основными средствами разработки пользовательских приложений для систем геометрического моделирования. Для достижения поставленной цели предлагается разработать процедуры добавления компонента, определенного во внешнем файле, в текущий документ.

Приложение, обеспечивающее программную поддержку конструкторского проектирования некоторого класса технических объектов в среде системы геометрического моделирования, может быть организовано как некая макрокоманда, включающая обращения к штатным процедурам и функциям пакета и являющаяся своего рода имитатором действий проектировщика. Реализовать такое приложение позволяет так называемый интерфейс программирования приложений (**Application Programming Interface, API**). Досконально его изучить можно при помощи соответствующей документации (справочник API). Повысить эффективность как изучения, так и непосредственно написания приложений позволяет макрорекордер – средство автоматической записи действий в формате какого-либо

языка программирования. API и макрорекордер – основные средства разработки приложений.

2.1. Начало работы

В качестве базовой системы геометрического моделирования предлагается SolidWorks. В качестве тестового примера – модель, созданная при выполнении работы «**Проектирование приспособления для установки и закрепления деталей**». Объект разработки – процедура, добавляющая в текущий документ компоненты, определенные во внешних файлах. Подобные операции выполняются на уровне сборки, поэтому для работы необходимо создать новый документ типа «сборка» (*Asmdoc*).

2.2. Макрокоманды

Чтобы выяснить, какие процедуры SolidWorks следует использовать, необходимо изучить справочник API или так называемый макрорекордер.

Команды управления макросами доступны в падающем меню: «**Инструменты** → **Макрос**». Кроме того можно включить соответствующую кнопочную панель.

1. Включить запись макроса.
2. Добавить в текущий документ какой-либо внешний компонент в интерактивном режиме – командой «Вставить компоненты» (панель инструментов «Сборка»).
3. Выключить запись макроса.

При выключении записи макроса система автоматически предложит сохранить его. Вариантов сохранения несколько. По умолчанию предлагается формат *Visual Basic for Application – SW VBA Macros(*.swp)*. Можно также сохранить макрос как проект *Visual Studio* в *Basic-формате* – *SW VSTA VB Macro(*.vbproj)*, либо формате «C#» – *SW VSTA C# Macro(*.csproj)*.

Далее рассматривается вариант *SW VBA Macros(*.swp)*.

2.3. Анализ и редактирование макроса

Файл, созданный на предыдущем этапе, открывается для редактирования. Сформированный модуль (рис. 2.1) содержит несколько

деклараций и процедуру *main*. Эта процедура и есть тот самый макрос, который был записан при выполнении вставки в интерактивном режиме.

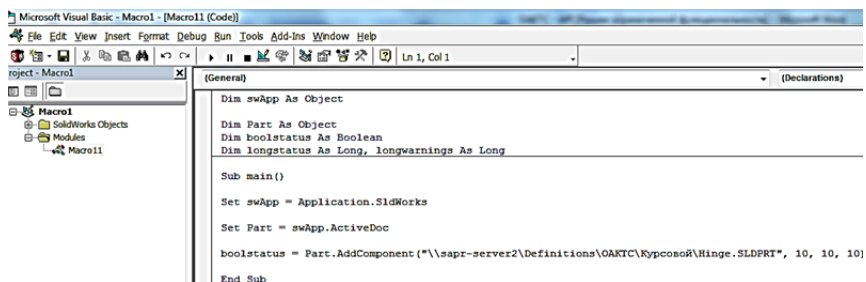


Рис. 2.1. Автоматически сформированный макрос

swApp – ссылка на приложение SolidWorks.

Part – ссылка на документ SolidWorks, в данном случае на текущий документ.

AddComponent – обращение к процедуре добавления детали из внешнего файла. Первый аргумент – полное имя файла, последующие три – координаты точки вставки.

Для проверки работоспособности этого макроса можно изменить координаты с последующим его исполнением. Убедиться в том, что деталь вставилась в файл сборки.

2.4. Справка по API

Процедуры и функции интерфейса программирования описаны в API-справочнике, загрузить который можно из раздела «Справка» падающего меню.

Наиболее эффективным является поиск по ключевым словам описания используемых процедур, названия которых можно взять из макросов, записываемых в процессе интерактивного выполнения.

Следует обратить внимание на то, что одноименные методы, как принято в объектно-ориентированном программировании, встречаются в разных классах. В частности, процедура *AddComponent* определена и в классе *IAssemblyDoc*, и в классе *IDragOperator*.

2.5. Процедуры открытия и закрытия файла компонента

Добавление внешнего компонента в системе SolidWorks имеет одну особенность. Для ее демонстрации систему следует полностью выгрузить, загрузить снова, создать новый сборочный документ и выполнить ранее созданный макрос. Он не работает. Точнее он работает, скорее всего, но не так, как раньше – компонент не добавляется.

Дело в том, что внешнее определение, прежде чем вставлять в текущий документ, необходимо инициализировать в текущем сеансе работы. При выполнении команды в интерактивном режиме система это делает, но в макрос не записывает. Поэтому его необходимо доработать самостоятельно. Простейший способ инициализации – открыть соответствующий файл и, соответственно, закрыть после добавления. Для этого в макрос следует добавить две строки обращения к методам *OpenDoc6* и *CloseDoc* приложения SolidWorks (рис. 2.2). С аргументами этих методов предлагается ознакомиться самостоятельно.

```
Sub main()  
  
Set swApp = Application.SldWorks  
  
Set Part = swApp.ActiveDoc  
  
swApp.OpenDoc6 "\\sapr-server2\Definitions\OAKTC\Курсовой\Hinge.SLDPRT", swDocPART, swOpenDocOp  
  
boolstatus = Part.AddComponent("\\sapr-server2\Definitions\OAKTC\Курсовой\Hinge.SLDPRT", 10, 10  
  
swApp.CloseDoc "\\sapr-server2\Definitions\OAKTC\Курсовой\Hinge.SLDPRT"  
  
End Sub
```

Рис. 2.2. Команды инициализации документа

2.6. Заключение

Методы и средства разработки приложений на базе системы геометрического моделирования SolidWorks можно также при помощи учебных пособий: «*Учебные пособия SolidWorks*» > «*Улучшение производительности*» > «*Интерфейс программирования приложений (API) SolidWorks*».

Как следует из вышесказанного, макросы можно записывать также как проекты среды Visual Studio на языке C#, в частности. Однако при этом часто возникают сбои. Проще, пожалуй, сразу создать проект в этой среде, организовав доступ к SolidWorks. Фрагмент такого проекта приведен ниже.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Runtime.InteropServices;
using SolidWorks.Interop.sldworks;
namespace ConsoleApplication1
{
class Program
{
staticvoid Main(string[] args)
{
object swApp = Marshal.GetActiveObject("sldworks.application");
ISldWorks Swap = swApp asISldWorks;
object Adoc = Swap.ActiveDoc;
AssemblyDoc Asm = Adoc asAssemblyDoc;
string UnitsLibPath = "D:\\BNTU\\OAK\\Units\\";
string uName = "Locator1.sldprt";
string Addun = UnitsLibPath + uName;
int Cerr = 0;
int Cwar = 0;
Swap.OpenDoc6(Addun, 1, 0, "", Cerr, Cwar);
Component2 Comp = Asm.AddComponent5(Addun, 0, "", true, "", 0.1, 0.1, 0.1);
Swap.CloseDoc(Addun);
}
}
}
```

3. АВТОМАТИЗАЦИЯ СОПРЯЖЕНИЙ

3.1. Цель и задачи

Одной из важнейших процедур автоматизации конструирования является позиционирование элементов конструкций. Можно безукоризненно подобрать все детали и узлы, рассчитать их параметры и т. д., но если каждый из них придется устанавливать интерактивно, ценность всего этого будет сведена на нет.

Положения элементов описываются координатным способом. В пространстве проектирования, где ведется сборка, определяется система координат – глобальная (ГСК). С каждым элементом конструкции связана собственная система координат (ССК) – локальная. В конечном итоге положение компонента описывается матрицей 4×4 :

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} & p_1 \\ c_{21} & c_{22} & c_{23} & p_2 \\ c_{31} & c_{32} & c_{33} & p_3 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

где c_{ij} – косинус угла между i -й осью ГСК и j -й осью ССК;

p_i – i -я координата точки начала ССК элемента в ГСК.

Определять все эти координаты и углы сложно. Современные средства геометрического моделирования поддерживают так называемую **ассоциативную сборку** – пользователь указывает пары сопрягаемых (ассоциируемых) друг с другом граней, а система сама рассчитывает необходимые параметры расположения. Отмеченную возможность можно использовать и программно. Используемые при этом методы и средства являются предметом изучения предлагаемой работы.

Цель работы: освоение методов и средств установления сопряжений между элементами геометрических моделей сборочных единиц.

Для достижения поставленной цели предлагается решить следующие задачи:

– выполнить сборку двух деталей в системе геометрического моделирования, используя сопряжения, и записать соответствующий макрос;

– проанализировать полученный макрос, выделить процедуру, непосредственно выполняющую сопряжение, изучить особенности ее выполнения, аргументы, способ передачи в нее сопрягаемых элементов;

– разработать процедуру сопряжения компонентов сборки.

В качестве примера рассматривается вставка в цилиндрическое отверстие установочного элемента класса «палец установочный». В конечном итоге должна быть разработана процедура, которая сделает это автоматически (рис. 3.1).

3.2. Создание макрокоманды

Необходимо создать новый документ типа «сборка» и вставить в него деталь с цилиндрическим отверстием и установочный палец.

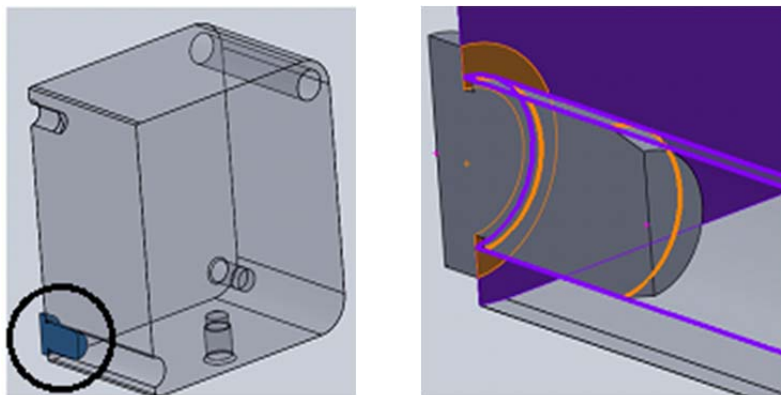


Рис. 3.1. Сопряжения установочного пальца с деталью

Положение пальца относительно детали регламентируется следующими сопряжениями (см. рис. 3.1):

– плоская грань буртика (прилегающая к цилиндрической) совпадает с торцевой гранью отверстия;

– цилиндрическая грань пальца соосна с отверстием.

Указанные сопряжения устанавливаются в заданном порядке. Макрос, записанный при интерактивном установлении указанных сопряжений, выглядит примерно следующим образом:

```
Sub main()  
Set swApp = Application.SldWorks  
Set Part = swApp.ActiveDoc  
boolstatus = Part.Extension.SelectByID2("", "FACE", 0.1, -0.0, -0.0,  
True, 1, Nothing, 0)  
boolstatus = Part.Extension.SelectByID2("", "FACE", 0.1, -0.0, 0.0,  
True, 1, Nothing, 0)  
Dim myMate As Object  
Set myMate = Part.AddMate3(0, 1, False, 0.01, 0, 0, 0.01, 0.01, 0, 0.5,  
0.5, False, longstatus)  
Part.ClearSelection2 True  
Part.EditRebuild3  
boolstatus = Part.Extension.SelectByID2("", "FACE", 0.1, -0.0, -0.0,  
True, 1, Nothing, 0)  
boolstatus = Part.Extension.SelectByID2("", "FACE", 0.1482, -0.0284,  
-0.0073, True, 1, Nothing, 0)  
Set myMate = Part.AddMate3(1, 1, False, 0.01, 0, 0, 0.01, 0.01, 0, 0.5,  
0.5, False, longstatus)  
Part.ClearSelection2 True  
Part.EditRebuild3  
End Sub
```

3.3. Анализ полученного макроса

Ключевая процедура в данном макросе – *AddMate3*. Именно она непосредственно устанавливает сопряжения.

Первый аргумент задает **тип сопряжения**. Это может быть «совпадение» сопрягаемых поверхностей, «соосность», «касательность», «параллельность» и т. п. В записанном макросе первое сопряжение – **совпадение** плоских поверхностей. Соответственно, первый аргумент принимает значение «0». Второе сопряжение – соосность (концентричность), первый аргумент равен «1». Очевидно, что тип сопряжения должен быть согласован с формой сопря-

гаемых поверхностей. Так, например, две плоские поверхности не сопрягаются «соосно».

Второй аргумент – **тип выравнивания**. Каждой поверхности геометрической модели можно поставить в соответствие вектор нормали или какой-либо характеристический вектор (например, ось). Тип выравнивания определяет взаимное расположение таких векторов сопрягаемых поверхностей. Если тип выравнивания равен «0», векторы будут сонаправлены, если значение «1» – направлены в противоположные стороны.

Тип выравнивания может принимать также значение «-1». В этом случае система сама определит тип выравнивания с учетом уже имеющихся сопряжений. Это полезно в случае сопряжения деталей по плоскостям и цилиндрическим поверхностям, как в приведенном выше примере. Направление нормали к плоскости очевидно, а направления векторов осей отверстия и пальца – нет. Соответственно, не известно, какой тип выравнивания следует указать во втором сопряжении, чтобы оно согласовывалось с сопряжением поверхностей. Поэтому в первую очередь следует сопрягать плоскости: тип выравнивания для них очевиден – «1». После этого можно сопрягать цилиндры с типом выравнивания «-1».

Остальные аргументы не имеют в данном контексте принципиального значения. С ними предлагается ознакомиться самостоятельно. Внимание следует обратить только на последний аргумент, значением которого в данном случае является переменная *longstatus*. В нее записывается целочисленный код ошибки, если таковая возникла при создании сопряжения.

В интерфейсе процедуры *AddMate3* отсутствуют аргументы, указывающие на сопрягаемые объекты. Каждому обращению к процедуре *AddMate3* предшествуют два обращения к процедуре *SelectByID2*. Процедура эта выделяет графический объект, со свойствами, соответствующими заданным значениям аргументов – имя, тип, координаты и др. То есть сопрягаемые объекты не передаются в процедуру сопряжения аргументами, а должны быть предварительно выделены.

3.4. Процедура сопряжения

Для ознакомления с особенностями сопряжения компонента сборки предлагается следующий вариант процедуры:

```

Sub MateLearning()
Set swApp = Application.SldWorks
Set Part = swApp.ActiveDoc
Dim myMate As Object
Dim longstatus As Long
Dim Tm As Long
Dim Ta As Long
Tm=0
Ta=1
Set myMate = Part.AddMate3(Tm, Ta, False, 0.0, 0.1, 0.1, 0.01, 0.001,
0, 0.5, 0.5, False, longstatus)
Part.ClearSelection2 True
Part.EditRebuild3
End Sub

```

Варьируя значениями переменных *Tm* и *Ta*, установить различные виды сопряжений. Сопрягаемые поверхности при этом выделяются предварительно до запуска процедуры.

При выполнении процедуры следует обратить внимание на зависимость между типом сопряжения и формой сопрягаемых поверхностей, а также на количество предварительно выделенных граней.

Установить последовательно два сопряжения² между одной и той же парой компонентов: совпадение плоскостей и концентричность, варьируя при этом типами их выравнивания, *Ta*.

4. ИЗВЛЕЧЕНИЕ ИНФОРМАЦИИ ИЗ ГЕОМЕТРИЧЕСКИХ МОДЕЛЕЙ В ТВЕРДОТЕЛЬНОМ ФОРМАТЕ

4.1. Цель работы и решаемые задачи

При решении многих конструкторских задач необходима геометрическая информация, которая уже содержится в модели в ее текущем состоянии. Анализ этой информации в ряде случаев позволяет

² Для этого можно дважды выполнить процедуру, предварительно выделяя каждый раз новую пару поверхностей. Можно также в одной процедуре сделать два сопрягающих фрагмента. При этом перед вторым обращением к методу *AddMate* определить точку паузы (*breakpoint*), во время которой выделить вторую пару граней.

избавить пользователя от избыточных операций ввода, повысив тем самым эффективность системы. Однако, прежде, чем анализировать, эту информацию необходимо извлечь из геометрической модели.

В настоящее время наиболее распространенной формой описания технических систем считается модель в формате **STEP** (STandard for EXchange of Product model data). Данный формат позволяет описать весь жизненный цикл изделия, включая технологию изготовления и контроль качества продукции. И, что самое важное, он поддерживает полноценное представление геометрии объекта – так называемую, твердотельную модель (SolidModel). Твердотельное моделирование поддерживается также всеми современными геометрическими системами, используемыми при конструировании.

Цель работы: освоение методов и средств доступа к элементам твердотельных геометрических моделей и извлечения доступной информации. Для достижения поставленной цели предлагается решить следующие задачи:

- ознакомиться с общей структурой твердотельной модели;
- изучить средства доступа к элементам этой структуры;
- разработать программные средства выполнения конструкторских процедур, использующих геометрическую информацию.

Средства поиска и анализа граней предлагается изучать на примере вставки внешнего компонента – цилиндрического пальца с буртиком, *LocatingPin*, в сборочную модель, содержащую деталь с отверстием, *TestAssm*.

Предполагается, что документ *TestAssm* открыт и активен. Пользователь указывает в модели *TestAssm* две грани для сопряжения и запускает процедуру, которая вставляет *LocatingPin* в *TestAssm* и устанавливает **два сопряжения** между элементом и моделью – совпадение плоскостей и концентричность цилиндров (рис. 4.1).

4.2. Твердотельная модель

Для конструкторского проектирования и его автоматизации наибольший интерес представляют две части твердотельной модели: **конструктивная геометрия и граничное представление.**

Конструктивная геометрия (*constructive solid geometry – CSG*) – множество объемных примитивов, соединенных бинарными теоретико-множественными операциями. **Примитив** (*Entity*) – элемен-

тарный объект. Соответственно, объемный примитив может пониматься как геометрическая простейшая фигура – призма, цилиндр, шар и т. п. Обобщение этого – тело, построенное **протягиванием** плоского **контура** S вдоль **траектории** T^3 (рис. 4.2).

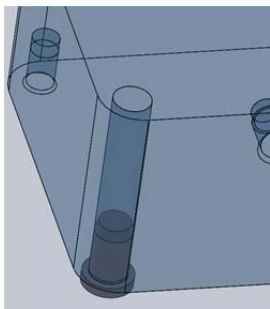


Рис. 4.1. Граничное представление детали

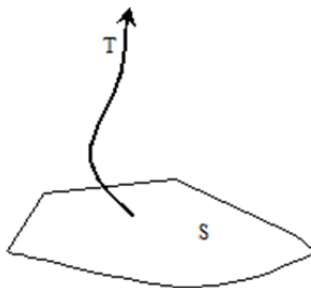


Рис. 4.2. Схема построения объемного примитива

Теоретико-множественные операции: **объединение** (*union*), **пересечение** (*intersection*), **вычитание** (*subtraction*). То есть, примитивы объединяются между собой, вычитаются один из другого либо пересекаются. Таким образом, сложные тела деталей получаются комбинированием объемных примитивов.

Кроме объемных примитивов конструктивная геометрия включает такие объекты, как системы координат, вспомогательную (или справочную, *reference*) геометрию и т. п. В каждом документе SolidWorks, например, присутствуют три взаимно перпендикулярные плоскости «сверху» (*top*), «справа» (*right*), «спереди» (*front*), а также объект «исходная точка» (*origin*). Эти объекты определяют систему координат, с одной стороны, а с другой – каждый из них является справочной (*reference*) плоскостью (точкой), используемой для геометрических построений. Пользователь может также сам создавать аналогичные вспомогательные объекты.

³ В геометрических системах определен более широкий набор способов построения и, соответственно, больше классов объемных примитивов: «Выдавливание» (*Extrusion*), «Вращение» (*Revolved*) и т. д. Однако эти операции сводятся к контуру и траектории: в первом случае траектория – вектор, во втором – дуга (спин-вектор).

Выражения, описывающие параметры модели, зависимости между размерами, связи размеров с параметрами, также относятся к конструктивной геометрии.

Таким образом, конструктивная геометрия отражает последовательность построения геометрической модели, ее структуру и способ построения. В системах геометрического моделирования конструктивная геометрия отображается в виде древовидной структуры (*Feature manager design tree*) рядом с графическим полем.

Конструктивная геометрия создается пользователем. Каждый ее элемент является уникальным свойством и имеет собственное имя⁴. По этому имени свойство может быть найдено в структуре модели.

Граничное представление (***Boundary REPresentation – BREP***) – метод представления объемной формы путем описания ее **границ**.

Границы **тела** – это **грани**, границы **граней** – это **ребра**, границы **ребер** – **вершины**. Очевидно, что смежные грани будут иметь общие ребра, смежные ребра – общие вершины. Подобную структуру удобно представлять в виде графа⁵:

$$G = (V, E),$$

где V – множество вершин, E – множество ребер. Например, V – множество граней, тогда E – множество ребер (то есть ребрам графа соответствуют ребра геометрической модели).

Граничное представление генерируется системой исходя из конструктивной геометрии. Элементы граничного представления не имеют индивидуальных отличительных признаков (например, собственных имен). Выделить их можно только посредством анализа их свойств – форм, размеров и др.

4.3. Доступ к элементам твердотельной модели

Под элементами твердотельной модели в данном контексте подразумеваются **свойства конструктивной геометрии и грани граничного представления**. Первые – это объекты класса *Feature*,

⁴ Имена эти генерируются системой и могут быть указаны проектировщиком.

⁵ Теория графов.

вторые – объекты класса *Face2*. И те, и другие доступны из-под компонента, содержащего этот элемент.

В рамках данной работы можно взять любой компонент. Полный перечень всех компонентов текущей сборки *Comps* возвращает метод *GetComponents*:

Comps = Asm.GetComponents(True),

где *Asm* – ссылка на объект класса *AssemblyDoc*⁶.

Перечень *Comps* возвращается в виде обычного одномерного массива, из которого выбирается любой элемент:

SetComp = Comps(i).

Переменная *Comp* должна быть объявлена как *Component2* или как *Object*.

Ссылка на свойство конструктивной геометрии может быть получена, в частности, методом *FeatureByName* непосредственно из компонента:

SetFeature = Comp.FeatureByName(<имя_свойства>).

Для получения доступа к элементам граничного представления необходимо сначала извлечь так называемое **тело** модели – объект класса *Body2* компонента:

SetCompBody = Comp.GetBody.

Body2 – это своего рода коллектор, в котором сосредоточена вся граничная геометрия – грани, ребра, вершины и др. При помощи метода *GetFaces* извлекается **массив** граней компонента:

CompFaces = CompBody.GetFaces.

Каждый из элементов возвращаемого массива – объект класса *Face2*.

Поскольку грани, как отмечалось ранее, собственных имен не имеют, в той или иной ситуации для выбора нужной следует анализировать их свойства.

Следует отметить, что часть свойств грани связана с поверхностью, участком которой эта грань является. Это, например, уравнения плоскости, оси и радиусы цилиндрических (конических) поверхностей и т. п. Эти свойства доступны из объекта класса *Surface*, возвращаемого методом *GetSurface* класса *Face*:

SetSurface = Face.GetSurface.

⁶ Может быть получена как ссылка на текущий документ, см. предыдущие работы.

Ознакомиться со свойствами граней и поверхностей предлагается самостоятельно, используя справочник по API системы SolidWorks.

4.4. Использование геометрической информации в процедурах

В качестве практического примера использования геометрических свойств объектов предлагается развитие процедуры сопряжений.

```
Sub main()  
Set swApp = Application.SldWorks  
Set Part = swApp.ActiveDoc  
boolstatus = Part.Extension.SelectByID2("", "FACE", 0.1172, -0.0246,  
-0.0042, True, 1, Nothing, 0)  
boolstatus = Part.Extension.SelectByID2("", "FACE", 0.1631, -0.0043,  
0.0613, True, 1, Nothing, 0)  
Dim myMate As Object  
Set myMate = Part.AddMate3(0, 1, False, 0.01, 0, 0, 0.01, 0.01, 0, 0.5,  
0.5, False, longstatus)  
Part.ClearSelection2 True  
Part.EditRebuild3  
boolstatus = Part.Extension.SelectByID2("", "FACE", 0.1711, -0.00752,  
-0.0126, True, 1, Nothing, 0)  
boolstatus = Part.Extension.SelectByID2("", "FACE", 0.1482, -0.0284,  
-0.0073, True, 1, Nothing, 0)  
Set myMate = Part.AddMate3(1, 1, False, 0.01, 0, 0, 0.01, 0.01, 0, 0.5,  
0.5, False, longstatus)  
Part.ClearSelection2 True  
Part.EditRebuild3  
End Sub
```

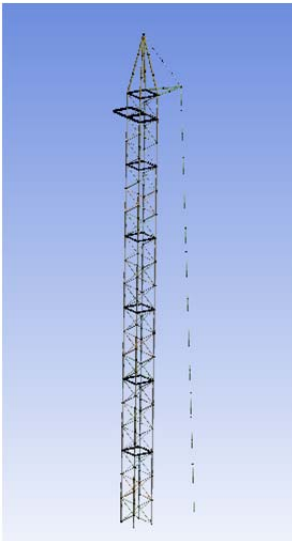
Сопрягаемые элементы попарно выделяются перед обращением к методу *AddMate3*. В приведенном фрагменте для выделения используется метод *SelectByID2* – интерактивное выделение посредством указания объекта в определенной точке, координаты которой передаются как аргументы. Вычислить точку на объекте в общем случае не просто. Кроме того, если объект известен, то нет нужды искать на нем точку, его можно выделить непосредственно методом *Select*.

```
Part.ClearSelection2 True
Object1ForMate.Select True
Object2ForMate.Select True
Set myMate = Part.AddMate3(Tm, Ta, False, 0.01, 0, 0, 0.01, 0.01,
0, 0.5, 0.5, False, longstatus)
```

Переменные **Object1ForMate** и **Object2ForMate** – ссылки на сопрягаемые элементы геометрических моделей. В данном качестве могут выступать элементы справочной (вспомогательной) геометрии либо грани. Первые являются элементами конструктивной геометрии, вторые – граничного представления. Доступ и к тем, и к другим рассматривался в разделе 0. Используя приведенный материал, предлагается установить различные варианты сопряжений, а также различные их сочетания.

Раздел 2

КОНЕЧНО-ЭЛЕМЕНТНОЕ МОДЕЛИРОВАНИЕ СЛОЖНЫХ СТЕРЖНЕВЫХ КОНСТРУКЦИЙ В ANSYSWORKBENCH



ВВЕДЕНИЕ

Балочные конструкции получили широкое распространение во всех направлениях строительства и являются распространенным конструктивным решением сооружений из металлоконструкций. Такая конструкция может состоять из трубчатых элементов, прокатных уголков, швеллеров и балок с другими сечениями. При моделировании балочной конструкции в конечно-элементной среде ANSYS Workbench 14.0 каждый стержень представлен элементом BEAM188 по умолчанию. В случае необходимости добавления тросов в конструкцию используется конечный элемент типа LINK180.

В качестве примера балочной конструкции взят башенный кран. Он используется на крупных стройплощадках и объектах, где требуются большие объемы строительных либо монтажных работ, скорость и высокая производительность.

1. ОПИСАНИЕ КОНСТРУКЦИИ



Рис. 1.1. Башенный кран КБ-403

Башенный кран КБ-403 (рис. 1.1) является самомонтирующимся грузоподъемным агрегатом с электрическим управлением, способным передвигаться на колесно-рельсовом ходу. Он оснащен подвижной балочной стрелой, закрепленной на такой же решетчатой поворотной башне, и управляемой посредством системы канатов.

Башня состоит из нескольких выдвижных секций с разборной головной секцией и возможностью подрачивания снизу. Балочная стрела имеет также секционную конструкцию.

Основные технические характеристики башенного крана:

- высота в свободном положении 37,9–54,7 м;
- база секций 1,8 м;
- количество секций башни до 5-ти штук;

- общая масса конструкции 76,6–80,5 т;
- масса противовеса: 30 т;
- вылет стрелы (минимальный/максимальный) 5,6/30 м;
- вылет стрелы при максимальном грузоподъеме 16,5 м;
- максимальная грузоподъемность 8 т;
- грузоподъемность при максимальном вылете стрелы 3 т;
- максимальный грузовой момент 120 т · м.

На кран, находящийся на строительной площадке, действуют силы и моменты со стороны груза, ветровые нагрузки, силы инерции движущихся частей, нагрузки от уклона, сила гравитации, которая зависит от собственной массы крана.

Проверка устойчивости крана производится в соответствии с нормами расчета по ГОСТ 13994-68 для двух случаев: при работе крана (грузовая устойчивость) и в нерабочем состоянии (собственная устойчивость).

Устойчивость крана определяют для наиболее неблагоприятных условий его работы. При проверке собственной устойчивости крана считают, что на него действуют ветровые нагрузки нерабочего состояния в сторону противовеса, кран стоит на уклоне (в сторону опрокидывания) без груза. При расчете грузовой устойчивости крана предполагают, что кран поднимает груз, равный его грузоподъемности на данном вылете, при этом груз имеет максимально возможную площадь; ветровые нагрузки рабочего состояния действуют со стороны противовеса, кран стоит на уклоне (в сторону груза). Для проверки устойчивости при внезапном снятии нагрузки считают, что кран располагается на уклоне в сторону опрокидывания, нагрузка на крюке принимается направленной вверх, а ветровая нагрузка рабочего состояния направлена со стороны стрелы.

Одна из особенностей расчета несущих конструкций башенных кранов заключается в определении деформаций и частот собственных колебаний. По японским нормам сейсмическая нагрузка учитывается в нерабочем состоянии при отсутствии ветра [6–8].

2. ПОСТАНОВКА ЗАДАЧИ

Цель работы: построить модель башни крана согласно заданным параметрам и произвести расчет напряженно-деформированного со-

стояния конструкции и грузовой устойчивости крана с учетом больших перемещений (рис. 2.2).

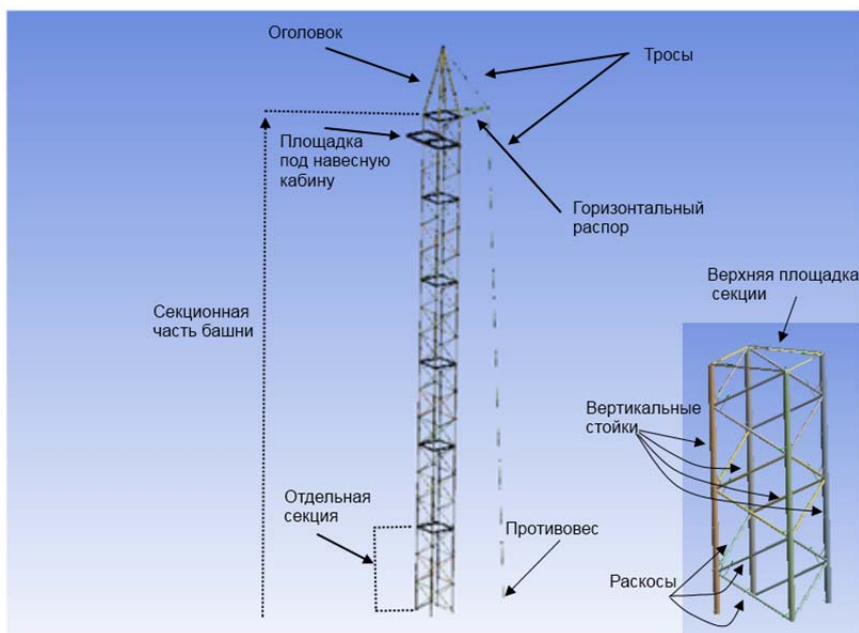


Рис. 2.2. Башенный кран КБ-403

Геометрическая часть модели башенного крана состоит из секционной части, оголовка башни, горизонтальной площадки для установки навесной кабины, верхнего распора и упрощенной системы тросов и полипласта.

Секционная часть включает в себя шесть секций. Каждая секция представляет собой балочную конструкцию высотой 6 м, длиной и шириной по 1,8 м. Секция состоит из четырех вертикальных стальных полых труб, в дальнейшем – стойки секции, диаметром 140 мм и толщиной стенки 8 мм, стальных полых труб раскосов диаметром 60 мм и толщиной стенки 5 мм. В верхней части секции имеется горизонтальная площадка, выполненная из равнополочных стальных уголков с высотой полки 50 мм и толщиной 6 мм.

В верхней части башни для установки навесной кабины на высоте 34 м имеется горизонтальная площадка, выполненная из стальных швеллеров высотой 100 мм, шириной полки 50 мм, толщиной 8 мм. Также в верхней части башни присутствует горизонтальный распор, расположенный на высоте 36 м и выполненный из стальных полых труб диаметром 140 мм и толщиной стенки 8 мм. Оголовок башни представляет собой рамную конструкцию из стальных полых труб диаметром 140 мм и толщиной стенки 8 мм. Система тросов и полиспаста упрощена и заменена двумя стальными тросами диаметром 40 мм, один из которых соединяет вершину оголовка башни с дальним концом распора, а второй соединяет дальний конец распора с противовесом.

Все компоненты выполнены из конструкционной стали, с модулем упругости 200 000 МПа ($2 \cdot 10^{11}$ н/м²) и коэффициентом Пуассона равным 0,33. Плотность материала равна 7850 кг/м³.

Нагрузки, приложенные к конструкции, включают вес самой башни, вес стрелы 2000 Н, вес груза 8000 Н, вес кабины 650 Н, силу от ветровой нагрузки 10 000 Н, действующей на башню со стороны противовеса, и силу от ветровой нагрузки 800 Н, действующей на груз.

В дальнейшем значения геометрических параметров переводятся в систему измерения СИ и все вычисления проводятся в системе измерения СИ.

3. ПОСТРОЕНИЕ ГЕОМЕТРИЧЕСКОЙ МОДЕЛИ БАШНИ КРАНА

Модель башенного крана имеет шесть одинаковых секций. Каждую секцию можно представить три раза повторяющейся частью.

Геометрическую модель конструкции будем строить следующим образом:

- построим повторяющуюся часть секции;
- построим одну секцию из трех повторяющихся частей секции;
- построим всю секционную часть башни (шесть секций);
- построим площадку под навесную кабину;
- построим оголовок башни;
- построим верхний распор башни;
- построим тросы башни.

3.1. Создание блока геометрии на схеме проекта

Для исследования конструкции используется пакет ANSYS 14.0. Запустите модуль Workbench 14.0:

Пуск > Все программы > ANSYS 14.0 > Workbench 14.0.

Открывшееся окно приложения содержит следующие элементы:

- MainMenu: главное меню приложения с пунктами File, Edit, View, Tools, Units, Help;
- Tool Menu: содержит пункты New, Open, Save, ...;
- Toolbox: окно слева, которое содержит разделы с доступными анализами AnalysisSystems, компонентами Component Systems, пользовательскими компонентами Custom Systems, компонентами оптимизации Design Exploration;
- ProjectSchematic: главное окно со схемой проекта (рис. 3.1).

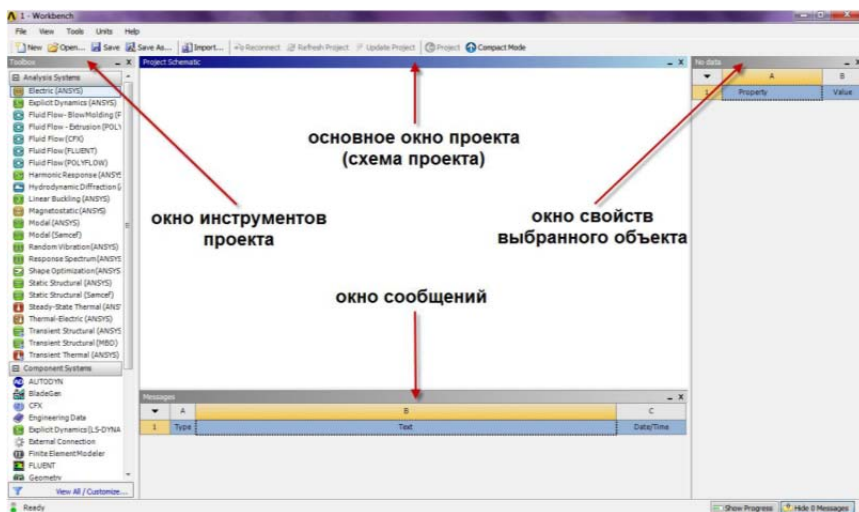


Рис. 3.1. Основное окно программы

При необходимости сбросьте настройки отображения окон, вернув их в положение по умолчанию:

{Main Menu} > View > Reset Window Layout.

Перетащите компонент *Geometry* в разделе *Toolbox > Component Systems* на окно со схемой проекта, зажав левую клавишу мыши

(отпустите клавишу мыши на том месте, где появится красная рамка). В схеме проекта добавится блок A: Geometry (рис. 3.2).

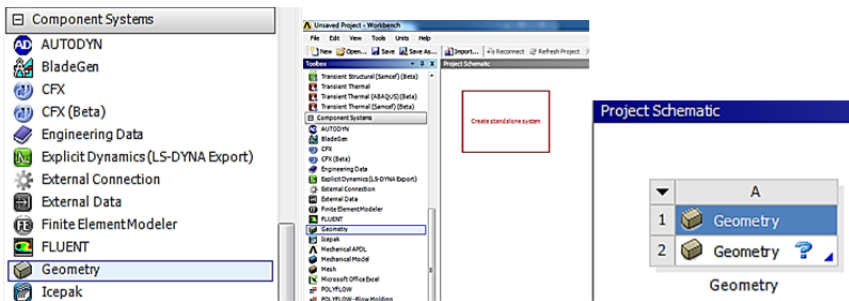


Рис. 3.2. Создание схемы проекта

Дважды щелкните по ячейке A2: *Geometry* со знаком вопроса. Запустится модуль для построения геометрии *DesignModeler*. Выберите метры в качестве единицы измерения (рис. 3.3).

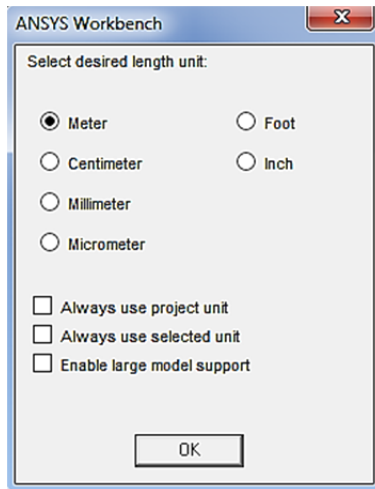


Рис. 3.3. Система единиц

Изучите *ToolMenu* модуля *DesignModeler*. Обратите внимание на следующие кнопки, отмеченные на рис. 3.4.

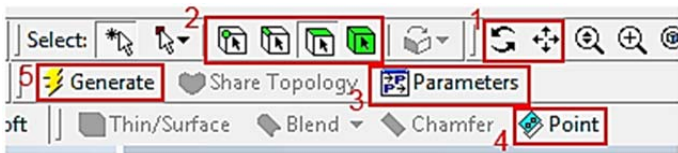


Рис. 3.4. ToolMenu модуля DesignModeler:

- 1 – поворот и перемещение конструкции в окне; 2 – ограничение выбора точки / линии / поверхности / объема; 3 – окно управления параметрами; 4 – создание точки; 5 – выполнение действия

При необходимости сбросьте настройки отображения окон, вернув их в положение по умолчанию:

{MainMenu} > View > Windows > ResetLayout.

3.2. Определение сечений стержней

Заранее создадим сечения для стержневых элементов конструкции башни.

Для создания сечения полой вертикальной трубы стойки выберите команду (рис. 3.5):

{MainMenu} > Concept > Cross Section > Circular Tube.

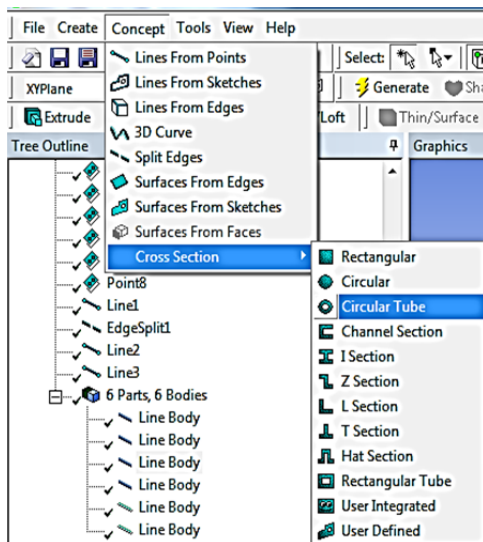


Рис. 3.5. Задание сечений

Задайте размеры согласно рис. 3.6. Для создания сечения полой трубы раскоса выберите снова команду:

{MainMenu} > Concept > Cross Section > Circular Tube.

Задайте размеры согласно рис. 3.7.

[-] Details of CircularTube1	
Sketch	CircularTube1
Show Constraints?	No
[-] Dimensions: 2	
<input type="checkbox"/> Ri	0,062 m
<input type="checkbox"/> Ro	0,07 m

Рис. 3.6. Размеры сечения трубы пояса

[-] Details of CircularTube2	
Sketch	CircularTube2
Show Constraints?	No
[-] Dimensions: 2	
<input type="checkbox"/> Ri	0,025 m
<input type="checkbox"/> Ro	0,03 m

Рис. 3.7. Размеры сечения трубы раскоса

Для создания сечения стержней верхней площадки секции в виде уголка выберите команду:

{MainMenu} > Concept > Cross Section > L Section.

Задайте размеры согласно рис. 3.8.

Для создания сечения стержней площадки под кабину в виде швеллера выберите команду:

{MainMenu} > Concept > Cross Section > Channel Section.

Задайте размеры согласно рис. 3.9.

[-] Details of L1	
Sketch	L1
Show Constraints?	No
[-] Dimensions: 4	
<input type="checkbox"/> W1	0,05 m
<input type="checkbox"/> W2	0,05 m
<input type="checkbox"/> t1	0,006 m
<input type="checkbox"/> t2	0,006 m

Рис. 3.8. Размеры сечения уголка площадки

[-] Details of Channel1	
Sketch	Channel1
Show Constraints?	No
[-] Dimensions: 6	
<input type="checkbox"/> W1	0,05 m
<input type="checkbox"/> W2	0,05 m
<input type="checkbox"/> W3	0,1 m
<input type="checkbox"/> t1	0,008 m
<input type="checkbox"/> t2	0,008 m
<input type="checkbox"/> t3	0,008 m

Рис. 3.9. Размеры сечения швеллера площадки кабины

Для создания сечения троса в виде круга выберите команду: *{MainMenu} > Concept > CrossSection > Circular*.
Задайте размеры согласно рис. 3.10.

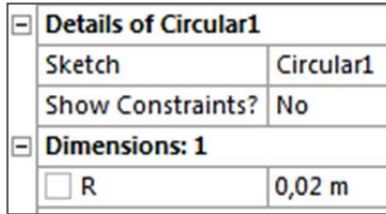


Рис. 3.10. Размеры сечения троса

Созданные сечения добавлены в узел *CrossSections* дерева модели (рис. 3.11).

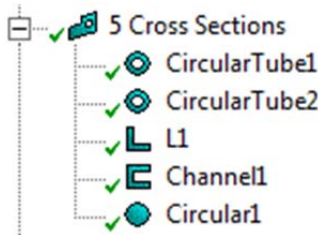



Рис. 3.11. Узел сечений

3.3. Создание параметрической геометрической модели

3.3.1. Построение отдельной секции

Заметим, что каждую секцию можно представить повторяющейся три раза частью. Если высоту секции принять за h , то высота повторяющейся части – $h/3$. Построение секции начнем с одной повторяющейся части секции. Полную отдельную секцию получим путем копирования данной части секции два раза.

Шаг 1: Создание основных параметров построения секции.

Нажмите на кнопку  **Parameters**, чтобы открыть окно управления параметрами *ParameterManager*. Данное окно содержит четыре вклад-

ки: **DesignParameters** (определение параметров для построения), **Parameter/DimensionAssignments** (присвоение значений геометрическим размерам), **Check** (проверка, подтверждение изменений), **Close** (закрытие окна). В первой вкладке **DesignParameters** задаем значения переменных (рис. 3.12):

$a = 0,9$ – половина ширины / длины секции;

$h = 6$ – высота секции;

$n = 5$ – количество повторов секций.

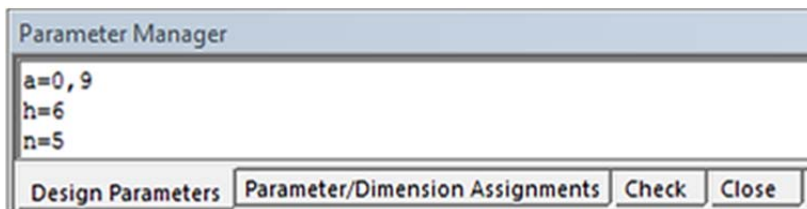



Рис. 3.12. Задание параметров

Шаг 2: Создание базовых точек повторяющейся части секции.

Нажмите на кнопку  **Point** (рис. 3.4), чтобы создать точку. Слева внизу откроется окно со свойствами. Поле **Type** содержит значение **Construction Point**, поле **Definition** – значение **Manual Input**.

В поле **FD8, XCoordinate** слева есть чекбокс, отметьте его. При этом в нем отобразится латинская буква **D**. Появится окно, предлагающее ввести имя параметра. Введите a , нажмите ОК (рис. 3.13).

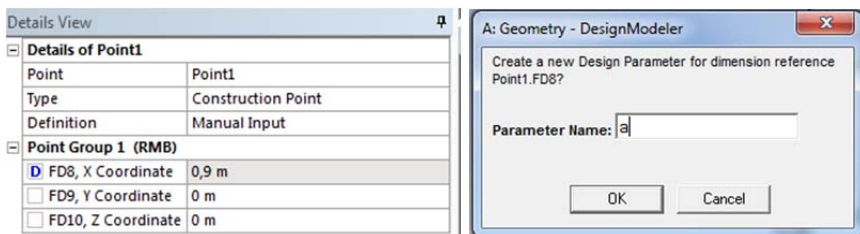


Рис. 3.13. Создание точки

Аналогичным образом в поле **FD9, YCoordinate** отметьте чекбокс, в окне введите a , нажмите ОК.

Нажмите кнопку  **Generate**. Построена точка Point 1 (рис. 3.14).

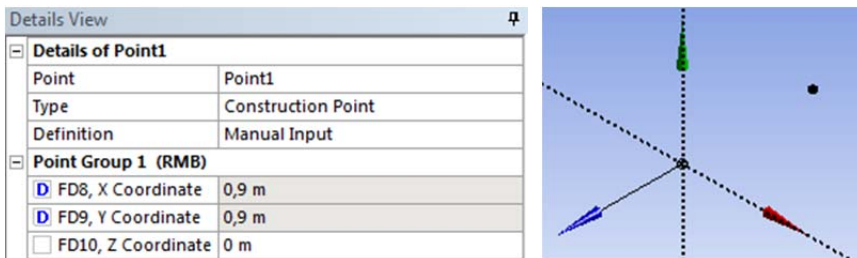


Рис. 3.14. Созданная точка Point 1

Вкладка *Parameter / Dimension Assignments* примет вид (рис. 3.15).

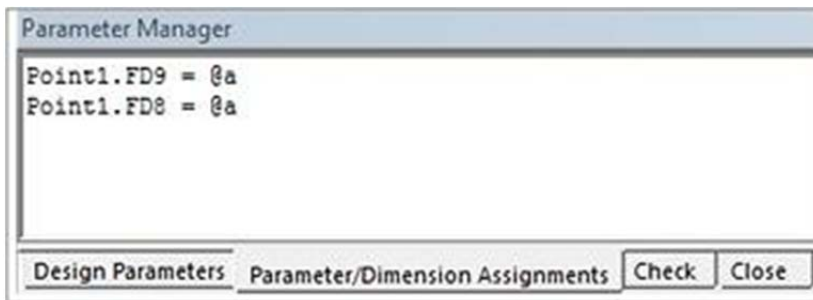




Рис. 3.15. Вкладка Parameter/Dimension Assignments

Type	Construction Point
Definition	Manual Input
Point Group 1 (RMB)	
<input type="checkbox"/> FD8, X Coordinate	0 m
<input type="checkbox"/> FD9, Y Coordinate	0 m
<input type="checkbox"/> FD10, Z Coordinate	0 m

Рис. 3.16. Свойства точки

Далее создайте еще семь точек со значениями координат по умолчанию. Нажмите на кнопку , чтобы создать очередную точку. Проверьте свойства (рис. 3.16). Нажмите кнопку  **Generate**. Повторите семь раз, чтобы получить восемь точек.

Откройте вкладку *Parameter / Dimension Assignments* и допишите значения геометрических параметров (координат) следующим образом (рис. 3.17).

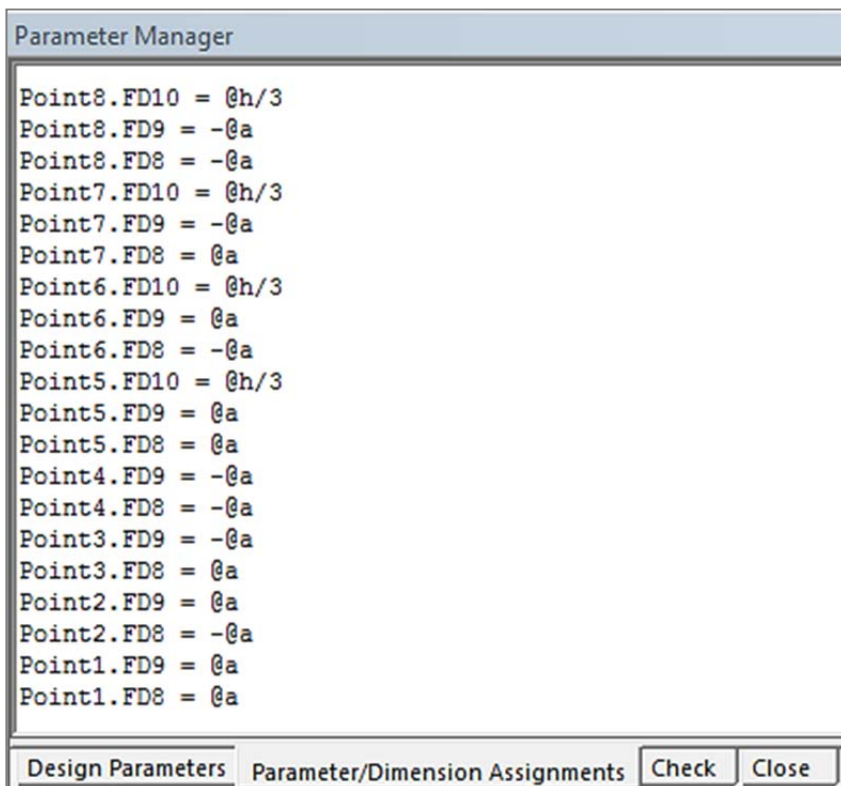


Рис. 3.17. Вкладка Parameter / Dimension Assignments

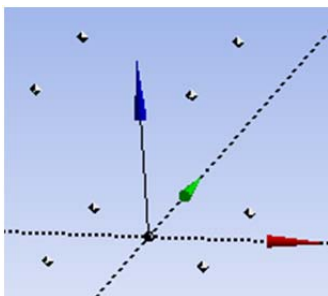



Рис. 3.18. Созданные точки

Обратите внимание, что для точек 5, 6, 7, 8 добавлен параметр **FD10** для координаты Z.

Нажмите на вкладку **Check**.
Нажмите кнопку  **Generate**.
Построенные точки отображены на рис. 3.18.

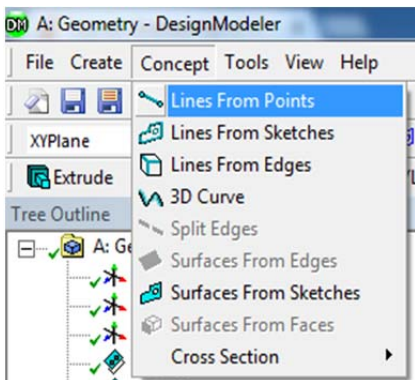



Рис. 3.19. Команда создания линии

В окне свойств нажмите *Apply* в поле *PointSegments* (рис. 3.20). Нажмите кнопку  *Generate*. Созданы четыре линии (рис. 3.21). Слева в дереве модели в узле *Parts*, *Bodies* появилось 4 *LineBody*. Переименуйте данные стержни из *LineBody* в *Base1*, *Base2*, *Base3*, *Base4* соответственно порядку (рис. 3.22).

Details of Line1	
Lines From Points	Line1
Point Segments	4
Operation	Add Material

Рис. 3.20. Свойства Line1

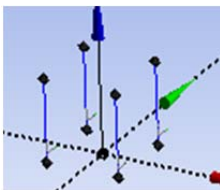


Рис. 3.21. Созданные линии



Рис. 3.22. Дерево модели

Далее необходимо разбить на две равные части две накрест лежащие линии. При этом делении в дерево модели не будет добавлено ни новых *LineBody*, ни новых *Point*, однако срединная точка и новые части деленной линии будут доступны для выделения.

Для разбиения нужных линий пополам выберите:
{MainMenu} > Concept > SplitEdges.

Выберите две линии, показанные на рис. 1.23, в окне свойств в поле *Edges* нажмите *Apply*. В поле *Definition* поставьте значение *Fractional*, в поле *FD1*, *Fraction* – значение *0,5* (рис. 1.24).

Шаг 3: Создание базовых линий повторяющейся части секции. Для создания линий по точкам выберите команду (рис. 3.19):
{MainMenu} > Concept > Lines FromPoints.

Постройте по созданным точкам следующие линии: щелчком мыши выберите точку 1, зажмите клавишу *Ctrl*, щелчком мыши выберите точку 5, отпустите *Ctrl*. Аналогично соедините точки 2–6, 3–7, 4–8.

Нажмите  **Generate**.

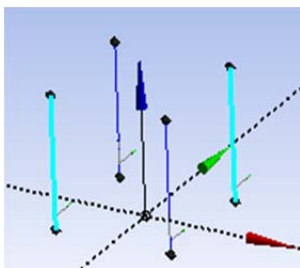



Рис. 3.23. Выбранные линии для разбиения пополам

Details of EdgeSplit1	
Line-Body Tool	EdgeSplit1
Edges	2
Definition	Fractional
<input type="checkbox"/> FD1, Fraction	0,5

Рис. 3.24. Свойства разбиения линии

Далее построим линии раскосов. Чтобы новые линии не объединились с уже построенными линиями в одно тело **LineBody** и было возможно задать линиям отличные сечения, необходимо при построении линий указывать операцию **AddFrozen**, а не **AddMaterial**.

Для построения линий раскосов по точкам выберите команду: **{MainMenu} > Concept > LinesFromPoints**.

Постройте линии, как показано на рис. 3.25. Поочередно соединяйте точки, не забывая удерживать клавишу **Ctrl**. Наметив все необходимые линии, в окне свойств в поле **PointSegments** нажмите **Apply**. В поле **Operation** выберите **AddFrozen** (рис. 3.26). Нажмите  **Generate**. В дереве модели в узле **Parts**, **Bodies** появилось новое тело **LineBody**, представляющее собой набор раскосов (рис. 3.27).

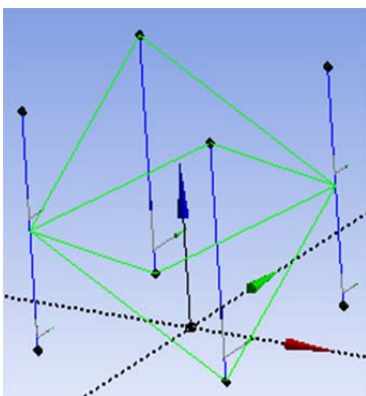


Рис. 3.25. Созданные линии раскосов

Details View	
[-] Details of Line2	
Lines From Points	Line2
Point Segments	8
Operation	Add Frozen

Рис. 3.26. Свойства линии

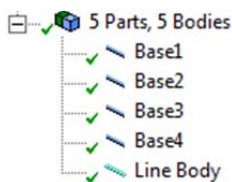


Рис. 3.27. Дерево модели

Шаг 4: Определение сечений стержней.

Припишем теперь построенным линиям нужные сечения. Выделите четыре вертикальные линии, представляющие стойки секции. В дереве модели это первые четыре узла **Base1**, **Base2**, **Base3**, **Base4**. В окне свойств в поле **CrossSection** выберите сечение **CircularTube1** (рис. 3.28).

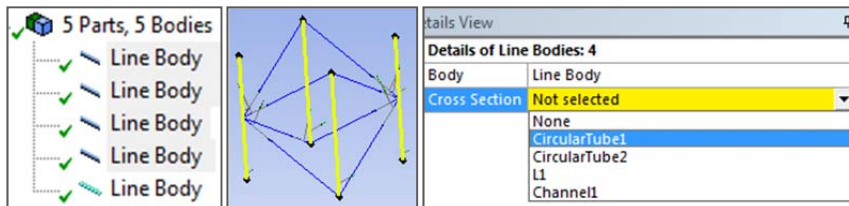


Рис. 3.28. Задание сечения вертикальным стойкам

Выделите оставшееся **LineBody** в дереве модели, которое представляет собой раскосы секции, и задайте для них в окне свойств в поле **CrossSection** сечение **CircularTube2**.

Шаг 5: Построение полной отдельной секции.

Полная секция состоит из трех повторяющихся частей.

Для копирования повторяющейся части секции выберите команду (рис. 3.29):

{MainMenu} > Create > Pattern.

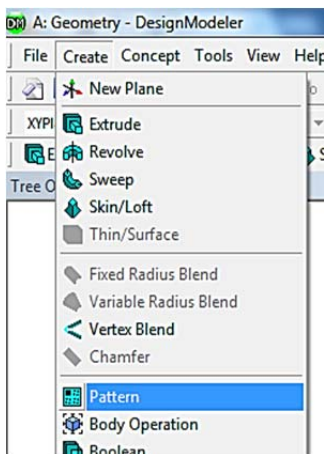


Рис. 3.29. Команда Pattern

Details of Pattern1	
Pattern	Pattern1
Pattern Type	Linear
Geometry	5 Bodies
Direction	2D Edge
<input checked="" type="checkbox"/> FD1, Offset	2 m
<input type="checkbox"/> FD3, Copies (>0)	2

Рис. 3.30. Свойства команды Pattern

В дереве модели выделите узел **ZXPlane** (рис. 3.31). Чтобы можно было выбрать ось *Z* в качестве направляющей для копирования, щелкните по пунктирной оси в графическом окне, соответствующей этой оси (рис. 3.32). Нажмите в окне свойств **Apply** в поле **Direction**.

В поле **FD1, Offset** (рис. 3.33) отметьте чекбокс, в окне введите *h*, нажмите ОК. Откройте вкладку **Parameter / DimensionAssignments** и измените значение параметра на $@h/3$ (рис. 3.33).

В поле **FD3, Copies** введите значение 2. Будет создано еще две копии построенной части сечения, т. е. всего *три* повторяющиеся части.

Нажмите  **Generate**.

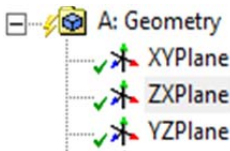


Рис. 3.31. Узел ZXPlane

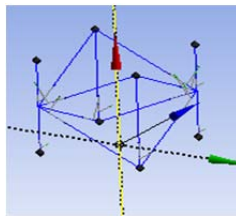


Рис. 3.32. Выбор направления

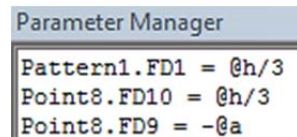


Рис. 3.33. Изменение параметра

Перейдем к построению горизонтальной площадки.

Выберите команду **{MainMenu} > Concept > LinesFromPoints** и создайте линии площадки, как показано на рис. 3.34, выбрав нужные точки с помощью мыши. Отметим все создаваемые линии. В окне свойств в поле **PointSegments** нажмите **Apply**. В поле **Operation** выберите **AddFrozen**.

Нажмите  **Generate**.

В окне свойств команды (рис. 3.30) проверьте значение **Linear** в поле **PatternType** – копирование линейно в заданном направлении.

Выделите все построенные тела сечения нажмите в окне свойств **Apply** в поле **Geometry**.

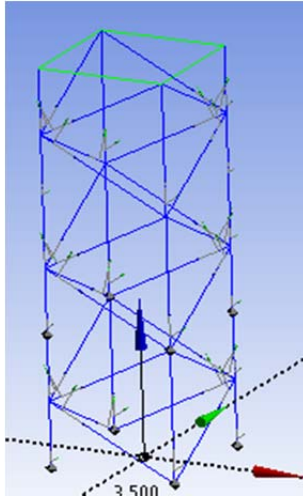


Рис. 3.34. Построение площадки

Выделите новое тело **LineBody** в дереве модели, которое представляет собой площадку, и задайте ему в окне свойств в поле **CrossSection** сечение **L1**.

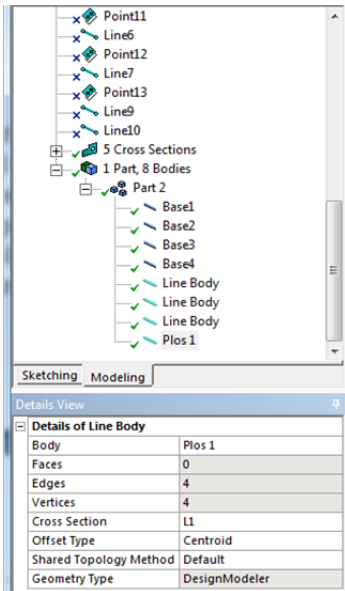



Рис. 3.35. Узел Plos1

Переименуйте тело в Plos1.

Для отображения на модели сечений стержней выберите пункт меню **{MainMenu} > View > Cross Section Solids**.

Проверьте, чтобы уголки площадки были ориентированы внутрь-вверх.

1. Нажмите левой кнопкой мыши в дереве модели по узлу Plos1 (рис. 3.35).

2. В данном теле Plos1 выделите отдельный стержень. Для этого нажмите кнопку , а затем нажмите на нужный стержень. Он подсветится зеленым цветом (рис. 3.36).

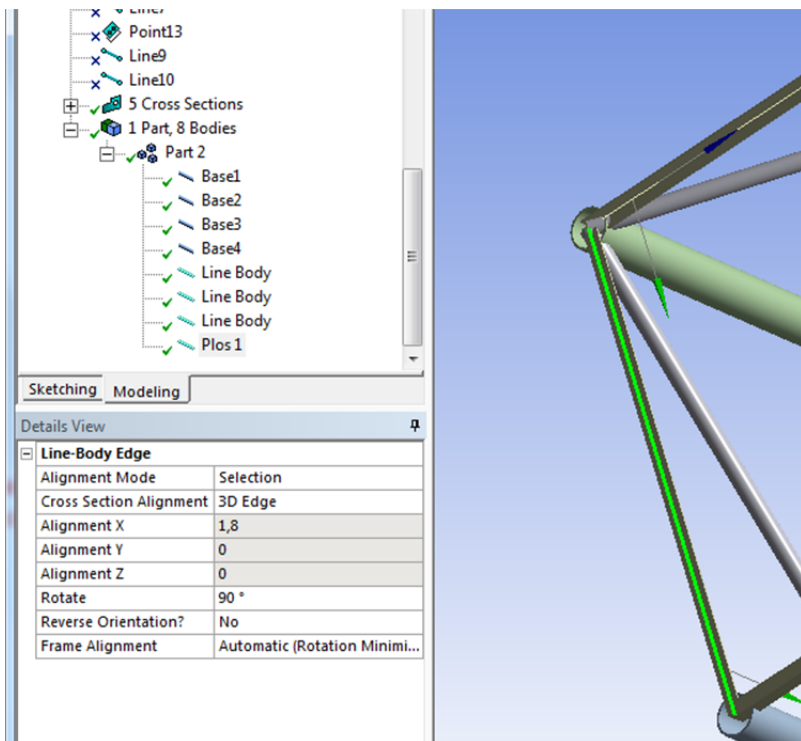


Рис. 3.36. Выбранный стержень

3. Выберите в выпадающем списке пункт "*Selection*" (а не "*Vector*") (рис. 3.37).

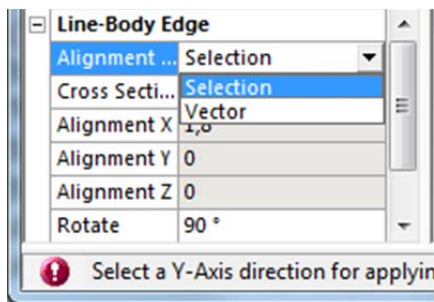



Рис. 3.37. Выбран пункт "Selection"

4. Нажмите кнопку , а затем нажмите на стержень, ортогональный по отношению к стержню, сечение которого следует повернуть (рис. 3.38). Он подсветится зеленым цветом.

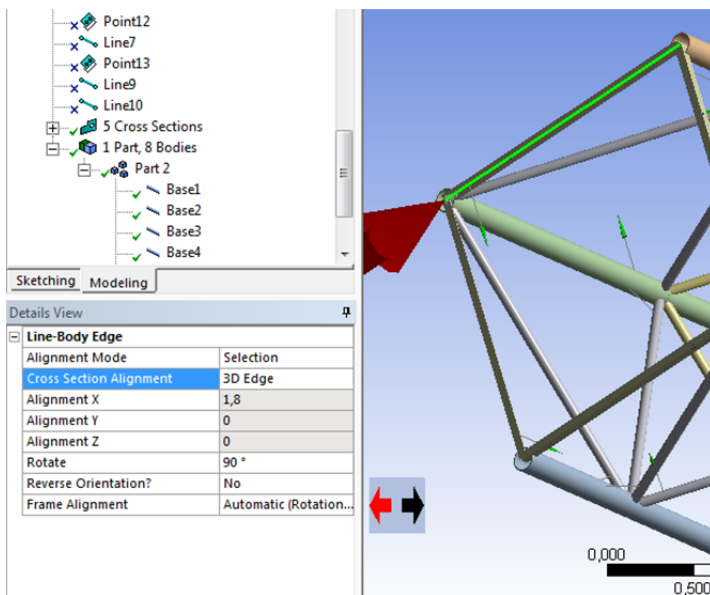


Рис. 3.38. Выбран стержень, ортогональный к поворачиваему

5. В появившемся окне свойств измените угол поворота в поле *Rotate* (рис. 3.39). Нажмите "Enter".

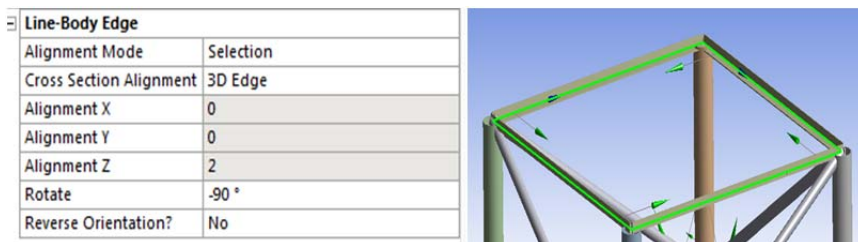


Рис. 3.39. Проверка ориентации сечения

3.3.2. Построение секционной части башни



Рис. 3.40. Секционная часть башни

Для построения секционной части башни выберите команду:

{MainMenu} > Create > Pattern.

Выделите все построенные тела секции и нажмите в окне свойств **Apply** в поле **Geometry**.

В дереве модели выделите узел **ZXPlane**, выберите ось **Z** в качестве направляющей, нажмите в окне свойств **Apply** в поле **Direction**.

В поле **FD1, Offset** отметьте чекбокс, в окне введите h , нажмите ОК.

В поле **FD3, Copies** отметьте чекбокс, в окне введите n , нажмите ОК.

Нажмите  **Generate**.

Результат построения представлен на рис. 3.40. Если графическое отображение башни долго не появляется, отключите отображение сечений стержней.

3.3.3. Построение площадки под навесную кабину

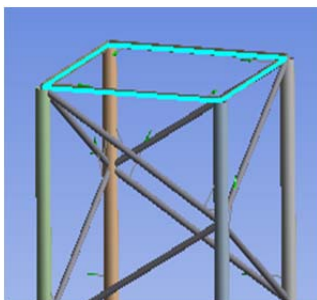



Рис. 3.41. Верхняя площадка башни

Для построения площадки под навесную кабину выберите команду:

{MainMenu} > Create > Pattern.

Выделите верхнюю площадку башни (рис. 3.41) и нажмите в окне свойств **Apply** в поле **Geometry**.

Выберите ось **Z** в качестве направляющей, измените направление копирования , нажмите в окне свойств **Apply** в поле **Direction**.

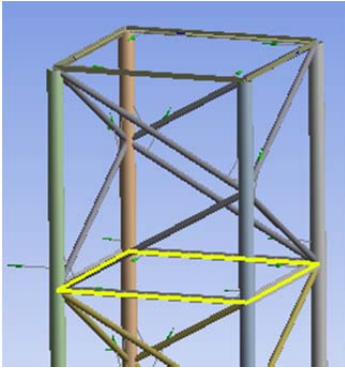


Рис. 3.42. Площадка под кабину

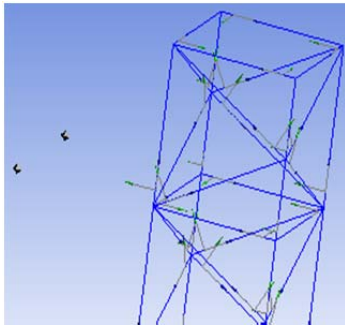


Рис. 3.43. Точки площадки под кабину

Выберите команду для построения линий:
{MainMenu} > Concept > LinesFromPoints.

Соедините точки согласно рис. 3.44.

В поле **Operation** выберите **AddFrozen**. Нажмите **Generate**.

Выберите в дереве проекта два тела, представляющие семь стержней площадки, задайте сечение **Channell** (рис. 3.45).

В поле **FD1, Offset** введите 2 (что соответствует копированию объекта двумя метрами ниже по оси Z).

В поле **FD3, Copies** введите 1.

Нажмите **Generate**. Полученная площадка отображена на рис. 3.42. Переименуйте полученное тело из четырех стержней из **LineBody** в **CabinFloor**.

Постройте две дополнительные точки площадки под кабину – **Point9** и **Point10**. Нажмите на **Point**, чтобы создать точку. Измените поле **Definition** – значение **ManualInput**. Нажмите **Generate**. Повторите.

Откройте вкладку **Parameter / DimensionAssignments** и измените значение параметра на:

$$Point10.FD10 = @h * (@n + 1) - @h/3$$

$$Point10.FD9 = @a + 2$$

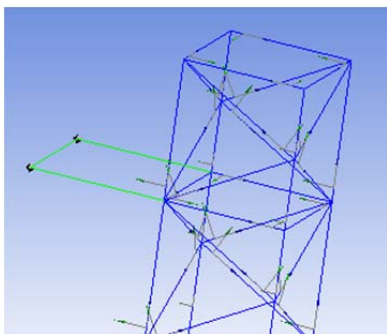
$$Point10.FD8 = -@a$$

$$Point9.FD10 = @h * (@n + 1) - @h/3$$

$$Point9.FD9 = @a + 2$$

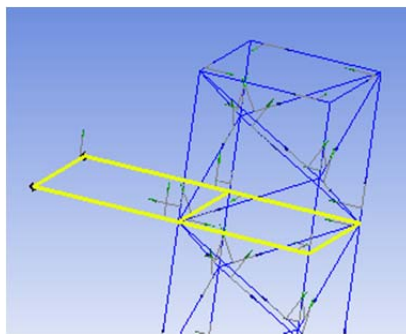
$$Point9.FD8 = @a$$

Нажмите **Generate**. Результат построения представлен на рис. 3.43.



Lines From Points	Line5
Point Segments	3
Operation	Add Frozen

Рис. 3.44. Площадка под кабину



Body	Line Body
Cross Section	Channel1

Рис. 3.45. Сечение площадки

Проверьте ориентацию сечений (рис. 3.46).

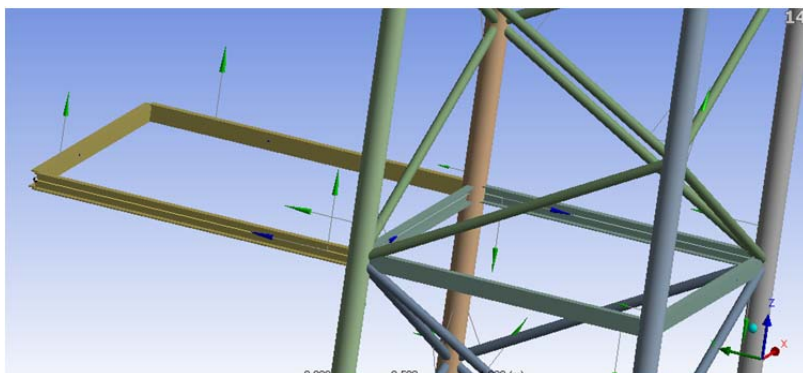




Рис. 3.46. Ориентация сечений площадки

Переименуйте полученное тело из трех стержней из **LineBody** в **Cabin**.

3.3.4. Построение распора баини

Постройте дополнительную точку распора – **Point11**. Нажмите на , измените поле **Definition** – значение **ManualInput**, нажмите

 **Generate**. Откройте вкладку *Parameter / DimensionAssignments* и измените значение параметра:

$$Point11.FD10 = @h * (@n + 1)$$

$$Point11.FD8 = @a + 3$$

Нажмите  **Generate**.

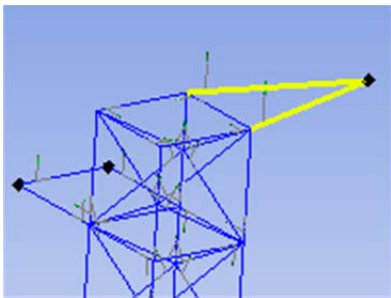



Рис. 3.47. Верхний распор башни

Выберите команду для построения линий:

{MainMenu} > *Concept* > *LinesFromPoints*.

Соедините необходимые точки согласно рис. 3.47. В поле **Operation** выберите *AddFrozen*. Нажмите  **Generate**.

Выберите стержни распора, задайте сечение *CircularTube1*.

Переименуйте полученное тело из *LineBody* в *Tale*.

3.3.5. Построение оголовка башни

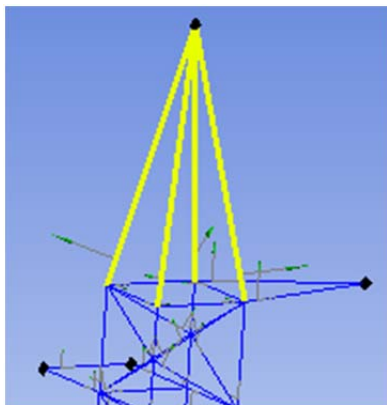





Рис. 3.48. Оголовок башни

Постройте дополнительную точку распора – *Point12*. Нажмите на  **Point**, измените поле **Definition** – значение *ManualInput*, нажмите  **Generate**. Откройте вкладку *Parameter / DimensionAssignments* и измените значение параметра:

$$Point12.FD10 = @h * (@n + 1) + 5$$

Нажмите  **Generate**.

Соедините необходимые точки согласно рис. 3.48. В поле **Operation** выберите *AddFrozen*. Нажмите  **Generate**.

Выберите четыре стержня оголовка, задайте сечение *CircularTube1*. Переименуйте полученное тело из *LineBody* в *Head*.

3.3.6. Построение тросов башни

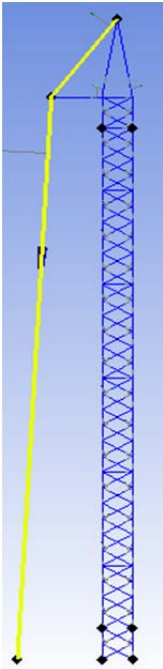


Рис. 3.49. Тросы башни

Постройте дополнительную точку противовеса – **Point13**.

Нажмите на **Point**, измените поле **Definition** – значение **ManualInput**, нажмите **Generate**.

Откройте вкладку **Parameter / Dimension Assignments** и измените значение параметра $Point13.FD8 = @a + 5$

Нажмите **Generate**.

Выберите команду для построения линий: **{MainMenu} > Concept > LinesFromPoints**.

Соедините точки **Point11** и **Point12**. В поле **Operation** выберите **AddFrozen**. Нажмите **Generate**.

Переименуйте полученное тело из **LineBody** в **CableHead**.

Снова выберите команду для построения линий. Соедините точки **Point12** и **Point13**. В поле **Operation** выберите **AddFrozen**. Нажмите **Generate**.

Переименуйте полученное тело из **LineBody** в **Cable**. Результат отображен на рис. 3.49.

3.3.7. Заключительный этап создания геометрии башни. Объединение всех тел в единую конструкцию

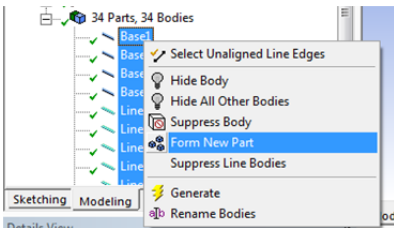


Рис. 3.50. Связывание тел

В дереве модели в узле **34 Parts, 34 Bodies** выберите все тела, щелкните правой клавишей мыши и выберите **FormNewPart**, чтобы при создании конечно-элементной сетки элементы имели общие узлы (рис. 3.50).

Закройте модуль **DesignModeler**.



4. ПОСТРОЕНИЕ КОНЕЧНО-ЭЛЕМЕНТНОЙ МОДЕЛИ БАШНИ КРАНА И ПРОВЕДЕНИЕ СТАТИЧЕСКОГО АНАЛИЗА КОНСТРУКЦИИ

4.1. Подготовка конечно-элементной модели башни

4.1.1. Создание блока анализа на схеме проекта

Перетащите компонент *StaticStructural* в разделе *Toolbox > AnalysisSystems* на ячейку *A2: Geometry* в схеме проекта. В схему проекта добавится блок *B: StaticStructural* (прил. 1).

Дважды щелкните по ячейке *B2: Engineering Data*. Откроется окно с материалами, доступными в проекте (рис. 4.1).

2		Material		
3	 Structural Steel	<input type="checkbox"/>		Fatigue Data at zero mean stress comes from 1998 ASME BPV Code, Section 8, Div 2, Table 5-110.1
*	Click here to add a new material			














Properties of Outline Row 3: Structural Steel				
	A	B	C	D E
1	Property	Value	Unit	 
2	 Density	7850	kg m ⁻³	<input type="checkbox"/> <input type="checkbox"/>
3	 Isotropic Secant Coefficient of Thermal Expansion			<input type="checkbox"/>
6	 Isotropic Elasticity			<input type="checkbox"/>
7	Derive from	Young's M...		
8	Young's Modulus	2E+11	Pa	<input type="checkbox"/>
9	Poisson's Ratio	0,3		<input type="checkbox"/>
10	Bulk Modulus	1,6667E+11	Pa	<input type="checkbox"/>
11	Shear Modulus	7,6923E+10	Pa	<input type="checkbox"/>
12	 Alternating Stress Mean Stress	 Tabular		<input type="checkbox"/> <input type="checkbox"/>
16	 Strain-Life Parameters			<input type="checkbox"/>
24	 Tensile Yield Strength	2,5E+08	Pa	<input type="checkbox"/> <input type="checkbox"/>
25	 Compressive Yield Strength	2,5E+08	Pa	<input type="checkbox"/> <input type="checkbox"/>
26	 Tensile Ultimate Strength	4,6E+08	Pa	<input type="checkbox"/> <input type="checkbox"/>
27	 Compressive Ultimate Strength	0	Pa	<input type="checkbox"/> <input type="checkbox"/>

Рис. 4.1. Материалы проекта

Просмотрите свойства материала по умолчанию – конструкционной стали. При необходимости можно добавить свой материал или

воспользоваться библиотекой материалов. Для возвращения к схеме проекта нажмите .

Дважды щелкните по ячейке *B4: Model* для открытия модуля *Mechanical*.

4.1.2. Учет навесной кабины на башне крана с помощью «точечной массы»

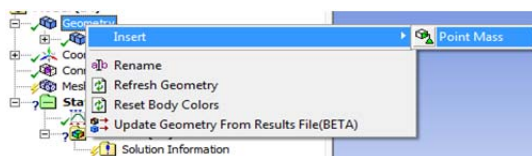
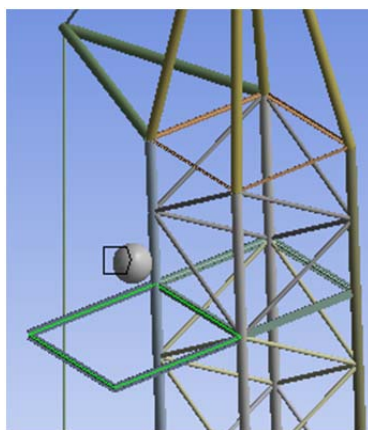


Рис. 4.2. Добавление массы

В дереве модели щелкните правой клавишей мыши по узлу *Geometry*. Выберите (рис. 4.2): *Insert > Point Mass*.

Выберите четыре стержня площадки под навесную кабину (из тела *Cabin* три стержня и из тела *CabinFloor* один стержень) и нажмите в окне свойств *Apply* в поле *Geometry* (рис. 4.3). Задайте координаты $X = 0$, $Y = 1,9$ м, $Z = 35$ м, которые соответствуют центру тяжести кабины. В поле *Mass* введите значение массы кабины *650*.

В данном варианте для упрощения не учитываются моменты инерции массы.



Scoping Method	Geometry Selection
Geometry	4 Edges
Coordinate System	Global Coordinate System
<input type="checkbox"/> X Coordinate	0, m
<input type="checkbox"/> Y Coordinate	1,9 m
<input type="checkbox"/> Z Coordinate	35, m
Location	Click to Change
Definition	
<input type="checkbox"/> Mass	650, kg
Mass Moment of Inertia X	0, kg·m ²
Mass Moment of Inertia Y	0, kg·m ²
Mass Moment of Inertia Z	0, kg·m ²

Рис. 4.3. Моделирование кабины точечной массой

4.1.3. Учет стрелы крана с помощью «удаленной точки»

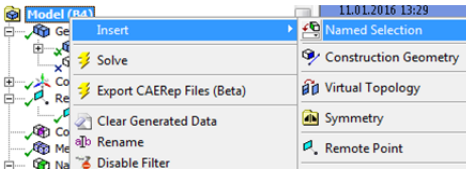
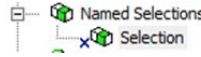


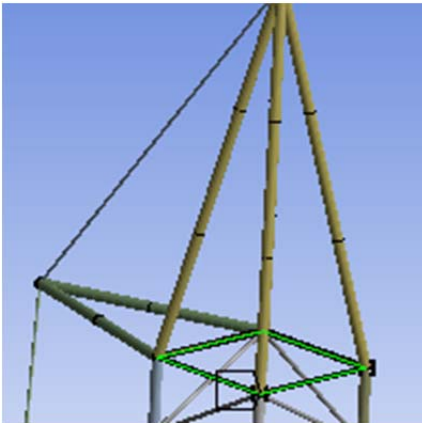
Рис. 4.4. Добавление выборки

В дереве модели щелкните правой клавишей мыши по главному узлу **Model (B4)**. Выберите (рис. 4.4):
Insert > NamedSelection.

В дерево модели добавлен узел



Выберите четыре стержня верхней площадки и нажмите в окне свойств **Apply** в поле **Geometry** (рис. 4.5) в окне свойств узла **Selection**.



Scope	
Scoping Method	Geometry Selection
Geometry	4 Edges
Definition	
Send to Solver	Yes
Visible	Yes
Program Controlled Inflation	Exclude
Statistics	
Type	Manual
Total Selection	4 Edges
Suppressed	0
Used by Mesh Worksheet	No

Рис. 4.5. Создание именованной выборки

Переименуйте узел **Selection** в **Arrow**

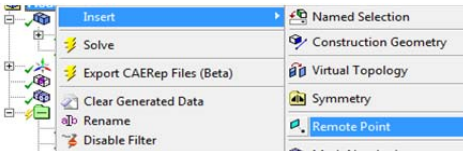
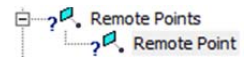


Рис. 4.6. Добавление удаленной точки

В дереве модели щелкните правой клавишей мыши по главному узлу **Model (B4)**. Выберите (рис. 4.6):
Insert > RemotePoint.

В дерево модели добавлен узел



Измените свойства удаленной точки согласно рис. 4.7. Переименуйте из **Remote Point** в **ArrowCenter**.

Scope	
Scoping Method	Named Selection
Named Selection	Arrow
Coordinate System	Global Coordinate System
<input checked="" type="checkbox"/> X Coordinate	-16,5 m
<input type="checkbox"/> Y Coordinate	0, m
<input type="checkbox"/> Z Coordinate	36, m
Location	Click to Change

Рис. 4.7. Создание удаленной точки

4.1.4. Учет ветровой нагрузки на кран

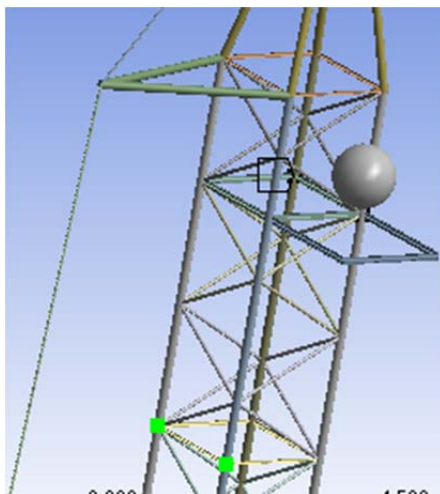
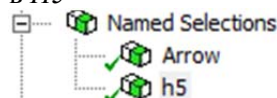


Рис4.8. Выбор точек ветровой нагрузки

В дереве модели щелкните правой клавишей мыши по узлу **NamedSelections**. Выберите **Insert > NamedSelection**.

Выберите две точки, расположенные со стороны противовеса на верхней площадке 5-й секции на высоте 30 м (рис. 4.8).

Переименуйте узел **Selection** в **H5**



4.1.5. Учет тросов крана

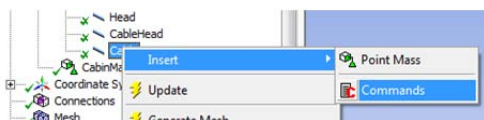


Рис. 4.9. Добавление свойств элемента

В дереве модели щелкните правой клавишей мыши по узлу **Geometry > Cable**. Выберите (рис. 4.9): **Insert > Commands**.

Трос моделируется при помощи конечного элемента типа LINK180.

Вставьте следующий код:

et,matid,180

r,matid,1.2566e-003,,

Величина $1.2566e-003$ равна площади сечения троса в m^2 . Данное значение можно увидеть, если выделить узел **Geometry > Cable** и в окне свойств просмотреть поле **CrossSectionArea**.

Выполните аналогичные действия для узла **Geometry > CableHead**. Код идентичен предыдущему.

При моделировании тросов стержень, представляющий трос, должен представляться в конечно-элементной сетке как *один* конечный элемент. Для этого в дереве модели щелкните правой клавишей мыши по узлу **Mesh**. Выберите **Insert > Sizing**. Выберите два стержня **Cable** и **CableHead**. Нажмите в окне свойств **Apply** в поле **Geometry**. В поле **Type** выберите значение **Number of Divisions**, в поле **Number of Divisions** введите **1** (рис. 4.10).

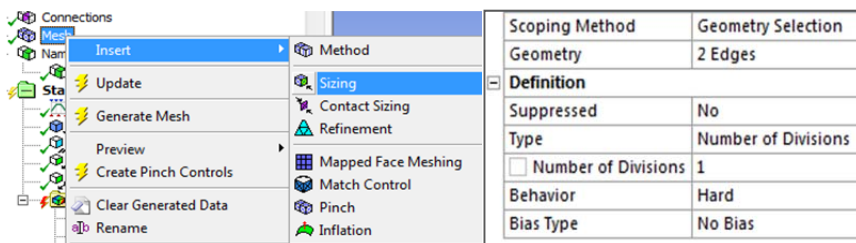


Рис. 4.10. Задание параметров для разбиения тросов на конечные элементы

В дереве модели щелкните правой клавишей мыши по узлу **Mesh** и выберите **GenerateMesh**.

При разбиении модели на конечные элементы, стальные стержни каркаса башни представлены по умолчанию элементами типа BEAM188, тросы – указанными нами для них элементами типа LINK180.

4.2. Задание граничных условий

4.2.1. Закрепление опор башины

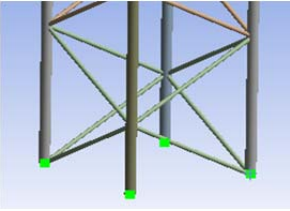



Рис. 4.11. Выбор опор

В дереве модели щелкните правой клавишей мыши по узлу **StaticStructural (B5)**, выберите **Insert > FixedSupport** (рис. 4.12).

Выберите 4 точки опоры (рис. 4.11), предварительно выбрав инструмент . Нажмите в окне свойств **Apply** в поле **Geometry**.

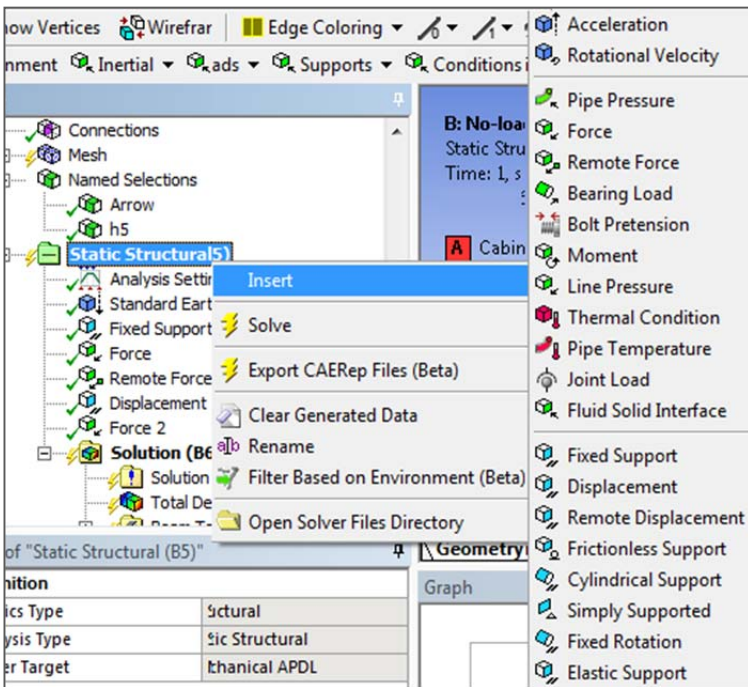


Рис. 4.12. Задание граничных условий

4.2.2. Собственный вес конструкции

Для учета собственного веса конструкции в дереве модели щелкните правой клавишей мыши по узлу **StaticStructural (B5)**, выберите

Insert > StandardEarthGravity. Проверьте правильность направления гравитационной силы ($-Z$).

4.2.3. Противовес

В дереве модели щелкните правой клавишей мыши по узлу **StaticStructural (B5)**, выберите **Insert > Displacement**. Выберите точку противовеса **Point13** в качестве точки приложения ограничения. В полях **XComponent**, **YComponent**, **ZComponent** поставьте 0 (рис. 4.13).

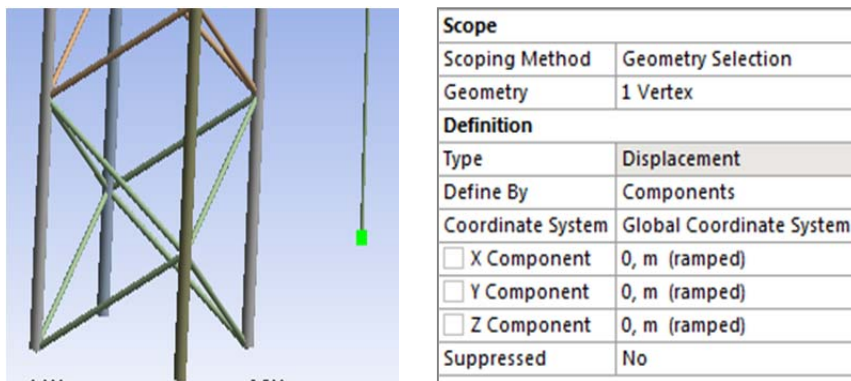


Рис. 4.13. Задание силы противовеса

4.2.4. Ветровая нагрузка

Нормальное ветровое давление на препятствие в первом приближении определяется по формуле

$$w_0 = 0,5 \cdot p \cdot v^2, \quad (1)$$

где v – скорость ветра, м/с, на высоте 10 м;

p – плотность воздуха, кг/м^3 , зависящая от его влажности, температуры и атмосферного давления;

$0,5$ – коэффициент сопротивления (обтекания).

Ветровую нагрузку рассчитываем для скорости ветра 15 м/с при плотности воздуха $1,2 \text{ кг/м}^3$. Учитываем изменение ветрового давления по высоте на 20 м; тип местности В – городские территории,

лесные массивы и другие местности, равномерно покрытые препятствиями высотой более 10 м [9–10].

Скорость порывов ветра содержит постоянную составляющую, среднее значение скорости ветра и переменную составляющую, которую можно представить как сумму гармонических составляющих:

$$V(t) = V_{cp}(t) + \sum_{i=1}^{\infty} V_{пор\ i} \cdot \sin(\omega_{ki} \cdot t), \quad (2)$$

где $V_{cp}(t)$ – сравнительно медленно меняющееся среднее значение скорости ветра, м/с;

$V_{пор\ i}$ – амплитуда i -й составляющей скорости порыва ветра;

ω_{ki} – частота i -й гармонической составляющей скорости порыва ветра.

Частота колебаний отдельных гармонических составляющих скорости в порыве ветра:

$$\omega_k = \pi \cdot K, \quad (3)$$

где K находится из эмпирического выражения [6]:

$$K = 0,0038 \cdot V_{cp}^{2,28} \text{ порывов/с.} \quad (4)$$

Для средней скорости 15 м/с частота $\omega_k = 5,73$ рад/с.

Scope	
Scoping Method	Named Selection
Named Selection	h5
Definition	
ID (Beta)	578
Type	Force
Define By	Components
Coordinate System	Global Coordinate System
<input type="checkbox"/> X Component	-20000 N (ramped)
<input type="checkbox"/> Y Component	0, N (ramped)
<input type="checkbox"/> Z Component	0, N (ramped)
Suppressed	No

Рис. 4.14. Ветровая нагрузка

В дереве модели щелкните правой клавишей мыши по узлу **StaticStructural (B5)**, выберите **Insert > Force**, задайте свойства, как показано на рис. 4.14.

Рассчитанная ветровая нагрузка округленно равна 10 000 Н и прикладывается к каждой из двух точек выборки *H5* в направлении, противоположном направлению оси *OX*. Так как прикладываемая нагрузка будет равномерно разделена по двум точкам выборки, увеличим значение силы до –20 000 Н.

4.2.5. Нагрузка от поднимаемого груза

В дереве модели щелкните правой клавишей мыши по узлу **StaticStructural (B5)**, выберите **Insert > RemoteForce**.

Scoping Method	Remote Point
Remote Points	ArrowCenter
Coordinate System	Global Coordinate System
<input type="checkbox"/> X Coordinate	-16,5 m
<input type="checkbox"/> Y Coordinate	0, m
<input type="checkbox"/> Z Coordinate	36, m
Location	Click to Change
Definition	
ID (Beta)	148
Type	Remote Force
Define By	Components
<input type="checkbox"/> X Component	-12000 N (ramped)
<input type="checkbox"/> Y Component	0, N (ramped)
<input type="checkbox"/> Z Component	-1,e+005 N (ramped)

Задайте свойства, как показано на рис. 4.15. Кран должен обеспечивать подъем массы 8 000 кг при вылете стрелы на 16,5 м. Учитывая массу груза, массу стрелы, значения удаленной силы в данной точке будут равны $-100\,000$ Н в направлении, противоположном оси *OZ*. Учитывая усилия от ветровой нагрузки на груз, значения удаленной силы в данной точке будут равны $-12\,000$ Н в направлении, противоположном оси *OX*.

Рис. 4.15. Усилия со стороны груза

4.3. Проведение статического анализа конструкции

Перед решением задачи добавьте результаты для просмотра. В дереве модели щелкните правой клавишей мыши по узлу **Solution (B6)**, выберите **Insert > Deformation > Total** (общие перемещения). Также можно добавить картины для продольной силы **Axial force**, поперечной силы **ShearForce**, изгибающего момента **TotalBendingMoment**, момента кручения **TorsionalMoment**.

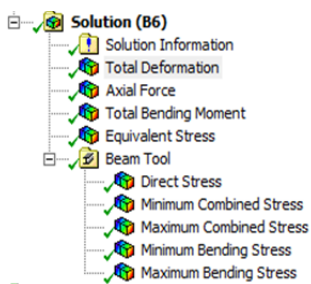


Рис. 4.16. Выполненный расчет

Insert > BeamTool > BeamTool. **BeamTool** включает картины для осевого напряжения **Directstress** и полного (сложного) напряжения **Combinedstress**.

Запустите расчет, щелкнув по кнопке **Solve**. Если расчет выполнен успешно, то возле добавленных результатов в узле **Solution (B6)** появится зеленая галочка (рис. 4.16).

Если в ходе расчета возникла ошибка, нажмите на узел **Solution-Information** и найдите информацию по возникшей ошибке.

На рис. 4.17 отображены суммарные перемещения узлов конструкции. Вершина башни имеет наибольшее перемещение, которое равняется 0,32 м.

На рис. 4.18 приведена картина максимальных комбинированных напряжений конструкции. При этом его наибольшее значение равняется 110 МПа (не превосходит предела текучести материала в 250 МПа), оно возникло в узле крепления распора. Наибольшее напряжение сжатия возникло в опоре башни **Base3** и равно -116 МПа, что также не превосходит предела текучести материала. Коэффициент запаса прочности по пределу текучести равен 2,16.

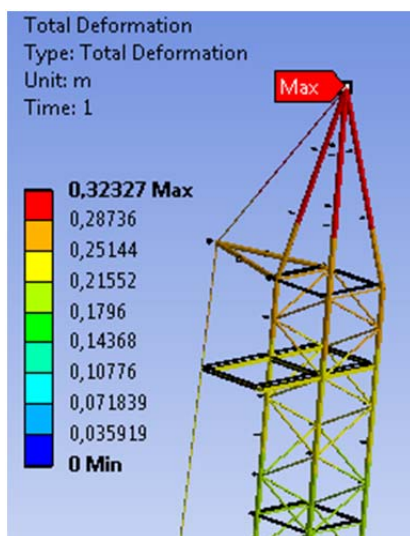


Рис. 4.17. Картина суммарных перемещений конструкции

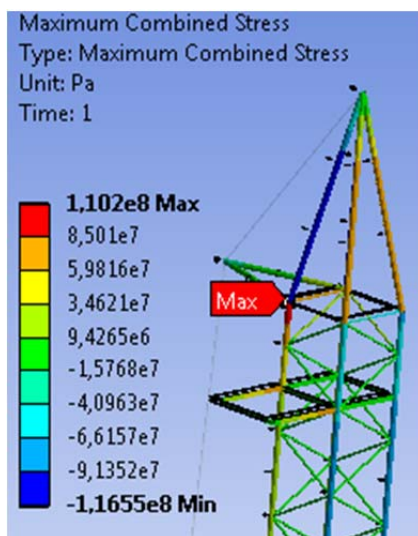


Рис. 4.18. Картина максимального полного напряжения конструкции

4.4. Анализ устойчивости конструкции

Откройте схему проекта. Перетащите компонент **LinearBuckling** из раздела **Toolbox > AnalysisSystems** на ячейку **B6: Solution** в блоке

B: StaticStructural на схеме проекта (рис. 4.19). В схему проекта добавится блок *C: LinearBuckling* (прил. 1).

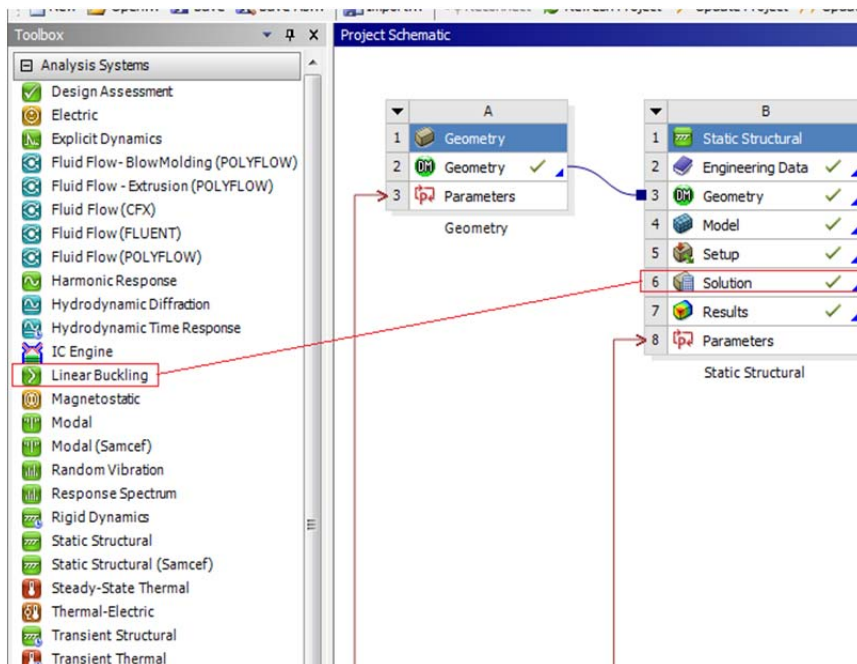


Рис. 4.19. Действие для создания нового анализа в проекте

Дважды щелкните по ячейке *C5: Setup* в блоке *C: LinearBuckling* для запуска модуля.

В настройках анализа *LinearBuckling* (*C5*) – *Analysis Settings* в поле *MaxModestoFind* можно задать число форм потери устойчивости. Для просмотра формы потери устойчивости в конце расчета добавьте общее перемещение элементов модели. В дереве модели найдите узел *LinearBuckling* (*C5*), щелкните правой клавишей мыши по узлу *Solution* (*C6*), выберите *Insert > Deformation > Total*.

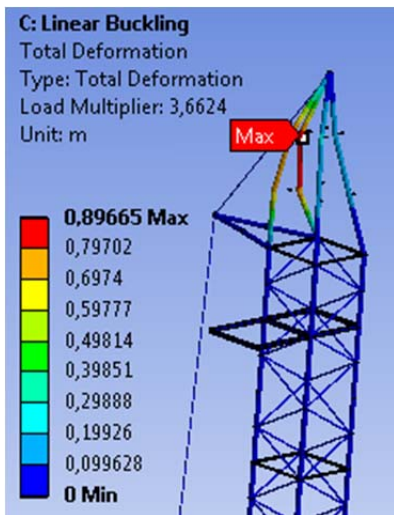


Рис. 4.20. Первая форма потери устойчивости

Запустите расчет: по узлу **LinearBuckling (C5) > Solution (C6)** щелкните правой клавишей мыши, выберите **Solve**.

После выполнения анализа выделяем в дереве узел **LinearBuckling (C5) > Solution (C6) > TotalDeformation**. В графическом окне отображается первая форма потери устойчивости модели (рис. 4.20). В табличных данных появляется параметр *LoadMultiplier*, равный 3,66. *LoadMultiplier* представляет собой коэффициент запаса устойчивости конструкции при потере устойчивости по первой форме.

4.5. Модальный анализ

Откройте схему проекта. Перетащите компонент из раздела **Toolbox > AnalysisSystems** на ячейку **B6: Solution** в блоке **B: Static-Structural** на схеме проекта. В схему проекта добавится блок **D: Modal** (прил. 1).

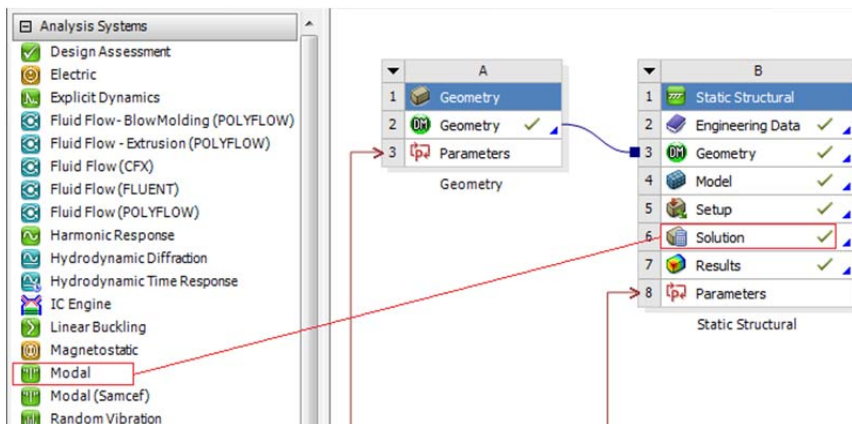


Рис. 4.21. Действие для создания нового анализа в проекте

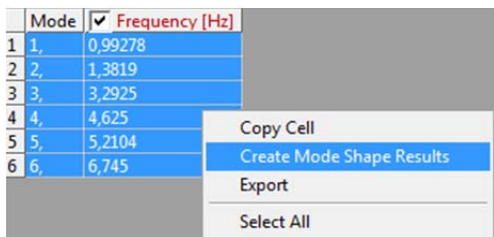
Дважды щелкните по ячейке *D5: Setup* в блоке *D: Modal* для запуска модуля.

В настройках анализа *Modal (D5) – AnalysisSettings* в поле *MaxModestoFind* можно задать число искомых мод собственных колебаний. Для просмотра собственной формы колебаний в дереве модели найдите узел *Modal (D5)*, щелкните правой клавишей мыши по узлу *Solution (D6)*, выберите *Insert > Deformation > Total*.

Запустите расчет: по узлу *Modal (D5) > Solution (D6)* щелкните правой клавишей мыши, выберите *Solve*.

После выполнения анализа выделяем в дереве узел *Modal (D5) > Solution (D6) > TotalDeformation*.

В графическом окне отображается первая форма собственных колебаний конструкции. В табличных данных появляются частоты первых шести мод.



Mode	Frequency [Hz]
1	0,99278
2	1,3819
3	3,2925
4	4,625
5	5,2104
6	6,745

Рис. 4.22. Отображение форм

Выделите все строки в таблице, щелкните правой клавишей мыши, выберите *CreateModeShapeResults* (рис. 4.22).

Получены следующие частоты собственных колебаний: для 1-й формы – 0,993 Гц; для 2-й формы –

1,382 Гц; для 3-й формы – 3,293 Гц; для 4-й формы – 4,625 Гц; для 5-й формы – 5,21 Гц; для 6-й формы – 6,745 Гц.

Обновите результаты: щелкните правой клавишей мыши по *Solution (D6)*, выберите *EvaluateAllResults*.

Первая частота собственных колебаний равна 0,993 Гц, а значит, ветра скоростью до 6 м/с не представляют опасности для конструкции, в чем можно убедиться, используя формулы (2)–(4) из раздела 4.2.4 в предположении одной гармонической составляющей. Частота колебаний отдельной гармонической составляющей скорости в порыве ветра для средней скорости 15 м/с равна $\omega_k = 5,73$ Гц.

Первая, пятая и шестая формы собственных колебаний представлены в формате *2xAuto* в прил. 2 на рис. П2.1–П2.3.

4.6. Гармонический анализ

Откройте схему проекта. Перетащите компонент **Harmonic** из раздела **Toolbox > AnalysisSystems** на ячейку **D6: Solution** в блоке **B: Modal** на схеме проекта (рис. 4.23). В схему добавится блок **E: HarmonicResponse** (прил. 1). Дважды щелкните по ячейке **E5: Setup** в блоке **E: HarmonicResponse** для запуска модуля.

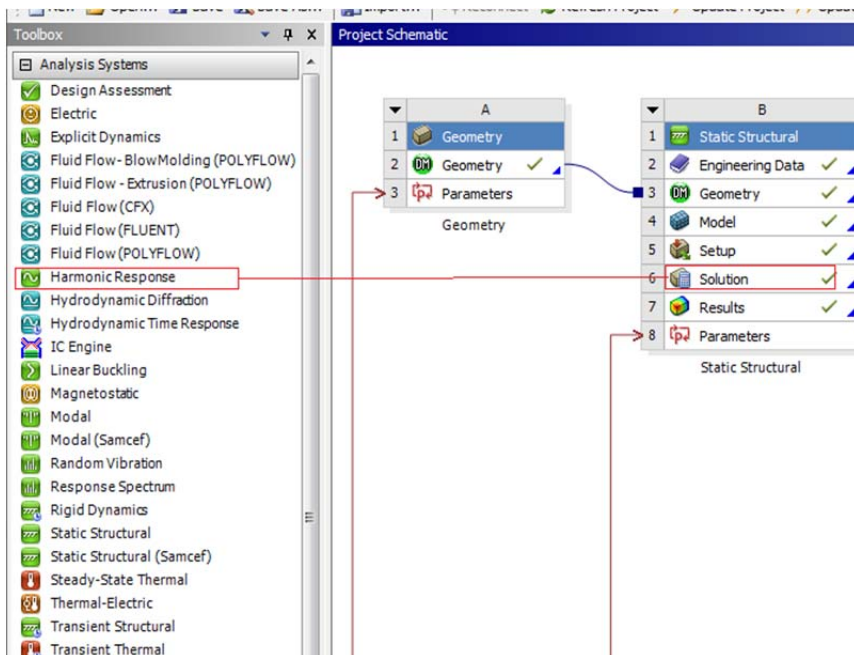


Рис. 4.23. Действие для создания нового анализа в проекте

В настройках анализа **HarmonicResponse (E5) – AnalysisSettings** в поле **RangeMinimum** – 0,6 Гц, в поле **RangeMaximum** – 6,8Гц, в поле **Solution Intervals** – 20.

В дереве модели щелкните правой клавишей мыши по узлу **HarmonicResponse (E5)**, выберите **Insert > Force**, задайте свойства, как показано на рис. 4.24.

Scoping Method	Named Selection
Named Selection	h5
Definition	
ID (Beta)	1056
Type	Force
Define By	Components
Coordinate System Global Coordinate System	
<input type="checkbox"/> X Component	-32000 N
<input type="checkbox"/> Y Component	0, N
<input type="checkbox"/> Z Component	0, N
<input type="checkbox"/> Phase Angle	0, °
Suppressed	No

Рис. 4.24. Задаем силу

Щелкните правой клавишей мыши по узлу **Solution (E6)**, выберите **Insert > FrequencyResponse > Acceleration**.

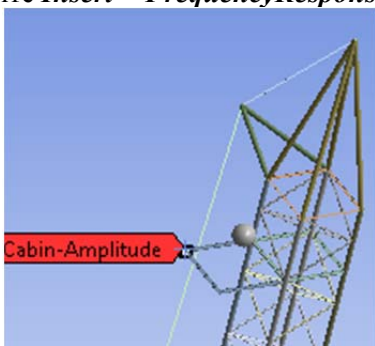


Рис. 4.25. Точка площадки

Амплитуда скорости ветра 25 м/с. Рассчитанная ветровая нагрузка округленно 16 000 Н и прикладывается к каждой из двух точек выборки *h5* в направлении, противоположном направлению оси *OX*. Так как прикладываемая нагрузка будет равномерно разделена по двум точкам выборки, увеличим значение силы до $-32\,000\text{ Н}$.

Выберите крайнюю дальнюю точку площадки под кабину конструкции (рис. 4.25) и задайте остальные настройки согласно рис. 4.26. Выберите **Insert > FrequencyResponse > Deformation**. Выберите крайнюю дальнюю точку площадки под кабину конструкции и задайте остальные настройки согласно рис. 4.27.

Scope	
Geometry	1 Vertex
Spatial Resolution	Use Average
Definition	
Type	Directional Acceleration
Orientation	X Axis
Suppressed	No
Options	
Frequency Range	Use Parent
Minimum Frequency	0,6 Hz
Maximum Frequency	6,8 Hz
Display	Amplitude

Рис. 4.26. Вид настроек

Scope	
Geometry	1 Vertex
Spatial Resolution	Use Average
Definition	
Type	Directional Deformation
Orientation	X Axis
Suppressed	No
Options	
Frequency Range	Use Parent
Minimum Frequency	0,6 Hz
Maximum Frequency	6,8 Hz
Display	Amplitude

Рис. 4.27. Вид настроек

Запустите расчет: по узлу *HarmonicResponse (E5)* > *Solution (E6)* щелкните правой клавишей мыши, выберите *Solve*. Просмотрите полученные результаты – графики *Frequency Response (Acceleration)* и *Frequency Response (Deformation)*.

На графике амплитуды деформации (рис. 4.28) можно заметить резонансные пики при частотах 1,53 Гц, 3,39 Гц, 5,25 Гц.

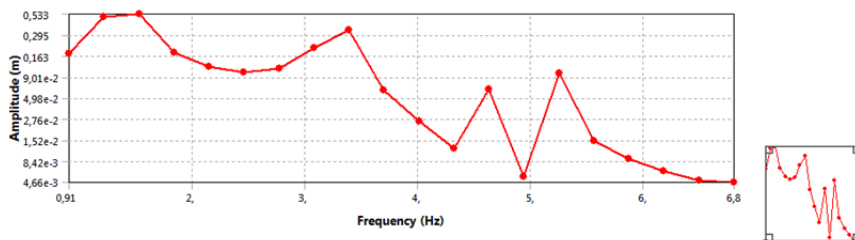


Рис. 4.28. График амплитуды деформации для точки площадки

Щелкните правой клавишей мыши по узлу *Solution (E6)*, выберите *Insert > Deformation > Total*. В окне свойств введите желаемую частоту колебаний, для которой должна быть отображена форма, например 1,53 Гц. Щелкните правой клавишей мыши по узлу *TotalDeformation* и выберите *RetrieveThisResult*.

Схема проекта

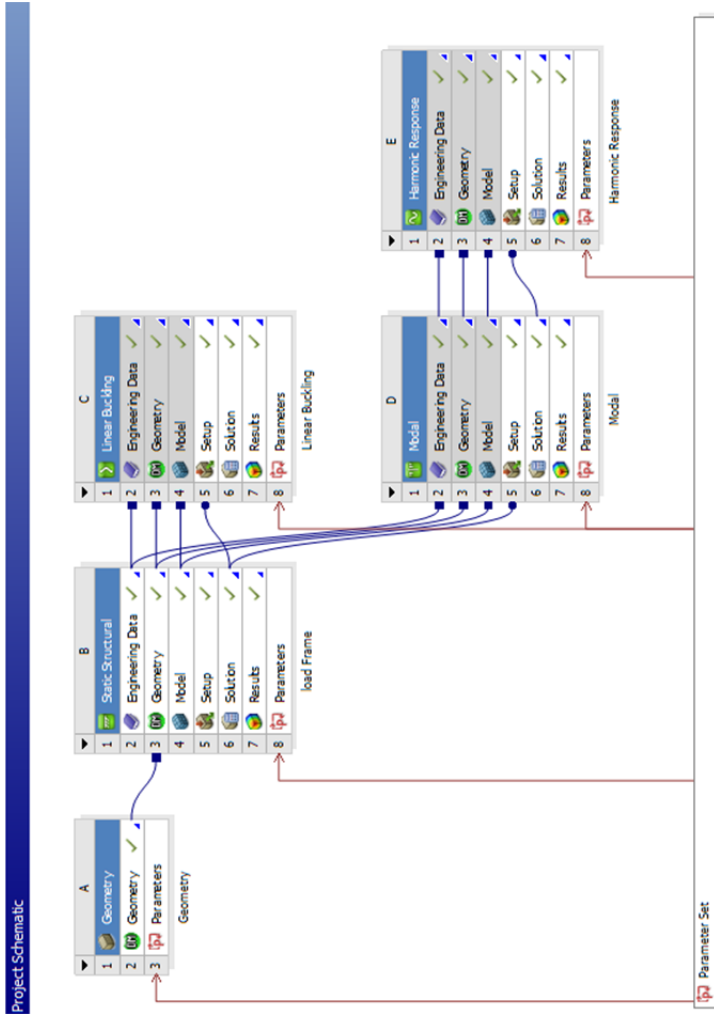


Рис. П.1.1. Схема проекта

Формы собственных колебаний

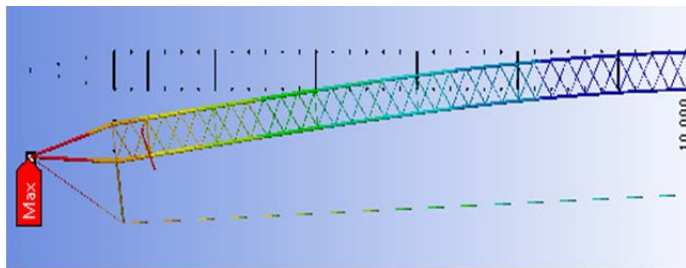


Рис. П2.1. Первая форма собственных колебаний (0,993 Гц)

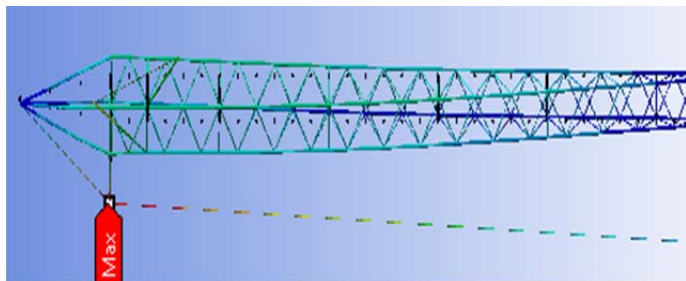


Рис. П2.2. Пятая форма собственных колебаний (5,21 Гц)

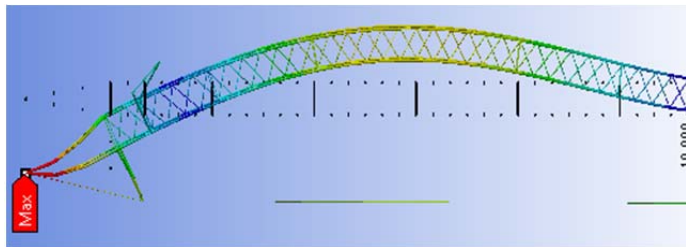


Рис. П2.3. Шестая форма собственных колебаний (6,745 Гц)

Задания по вариантам

Для видов секций, представленных на следующих рисунках, выполните посторенние конструкции крана, считая, что используется три таких секции. Место для крепления кабины выберите самостоятельно. Характеристики материала и нагрузки возьмите те же, что и в рассмотренном в пособии варианте. Выполните все виды расчетов и интерпретируйте результаты моделирования. Используя полученную модель, рассчитайте варианты с измененными толщинами раскосов и стоек. Выберите рациональный вариант по критерию материалоемкости.

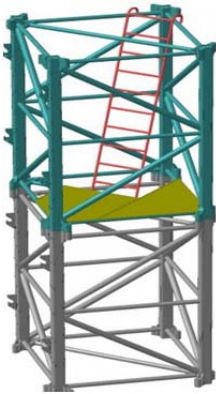


Рис. ПЗ.1. Вариант 1



Рис. ПЗ.2. Вариант 2



Рис. ПЗ.3. Вариант 3



Рис. ПЗ.4. Вариант 4

Раздел 3

ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ



ВВЕДЕНИЕ

Генетические алгоритмы (ГА) – это алгоритмы, которые позволяют найти удовлетворительное решение для аналитически неразрешимых или сложно решаемых проблем через последовательный подбор и комбинирование искомых параметров с использованием механизмов, напоминающих биологическую эволюцию.

ГА применяются для решения следующих задач:

- оптимизация функций;
- разнообразные задачи на графах (задача коммивояжера, раскраска, нахождение паросочетаний);
- настройка и обучение искусственной нейронной сети;
- задачи компоновки;
- составление расписаний;
- игровые стратегии;
- аппроксимация функций;
- искусственная жизнь;
- биоинформатика.

ГА обладают следующими преимуществами: они не требуют никакой дополнительной информации о поверхности функции выбора; разрывы, существующие в ней, незначительно влияют на эффективность оптимизации; устойчивы к попаданию в локальные оптимумы; хорошо работают при решении задач многоцелевой оптимизации; ГА просты в реализации, легко и удобно распараллеливаются.

Однако ГА обладают и существенными недостатками, например, с помощью ГА проблематично найти точный глобальный оптимум; ГА неэффективно применять в случае оптимизации функции, требующей большого времени на вычисление; не для всех задач удается найти оптимальное кодирование параметров.

С целью совершенствования ГА и преодоления указанных недостатков предлагается выполнить оптимизацию структур известных модификаций ГА.

1. ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

Цели работы:

– выполнить программную реализацию одной из модификаций классического генетического алгоритма,

- предусмотреть различные варианты реализации основных генетических операторов,
- сравнить эффективность полученных структур генетических алгоритмов.

2. КЛАССИЧЕСКИЙ ГА И ЕГО МОДИФИКАЦИИ

В настоящее время существует целый класс ГА, направленный на решение разнообразных задач [13–19]. ГА оперируют совокупностью особей (популяций), которые представляют собой строки, кодирующие одно из решений задачи. Этим ГА отличается от большинства других алгоритмов оптимизации, которые оперируют лишь с одним решением, улучшая его. С помощью функции приспособленности среди всех особей популяции выделяют:

- наиболее приспособленные (более подходящие решения), которые получают возможность скрещиваться и давать потомство;
- наихудшие (плохие решения), которые удаляются из популяции и не дают потомства.

Таким образом, приспособленность нового поколения в среднем выше предыдущего. Функция приспособленности (*fitness function*) должна принимать неотрицательные значения на ограниченной области определения (для того, чтобы мы могли для каждой особи считать ее приспособленность, которая не может быть отрицательной), при этом совершенно не требуются непрерывность и дифференцируемость. Особи, входящие в популяцию, в ГА представляются хромосомами с закодированными в них множествами параметров задачи, т. е. решений, которые иначе называются точками в пространстве поиска.

Примерами различных ГА могут являться алгоритмы, рассмотренные ниже.

2.1. Классический ГА

Классический ГА (называемый также стандартным, базовым, простым или элементарным) состоит из следующих шагов (рис. 2.1):

1. Инициализация, или формирование исходной популяции.
2. Вычисление функции приспособленности (вычисление пригодности) хромосом в популяции.
3. Вычисление условия окончания процесса.

4. Селекция хромосом с целью формирования пар родителей.

5. Применение генетического оператора скрещивания (*recombination*) хромосом промежуточной популяции путем кроссинговера или кроссовера (*crossover*), что приводит к формированию нового поколения.

6. Применение генетического оператора мутации (*mutation*).

7. Формирование новой популяции.

8. Формирование результата – выбор наилучшей хромосомы

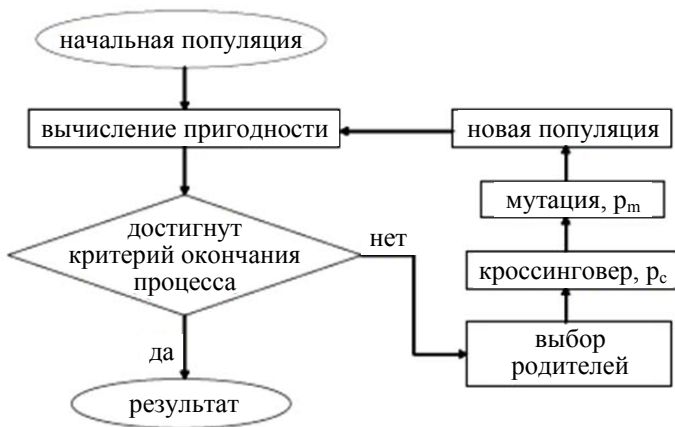


Рис. 2.1. Блок-схема классического генетического алгоритма

В классическом ГА:

- начальная популяция формируется случайным образом;
- размер популяции (количество особей N) фиксируется и не изменяется в течение работы всего алгоритма;
- используется двоичное представление хромосом, каждая особь генерируется как случайная L -битная строка, где L – длина кодировки особи;
- длина кодировки для всех особей одинакова;
- селекция выполняется методом рулетки;
- применяется одноточечное скрещивание.

Для повышения эффективности работы классического ГА создано множество его модификаций, в которых:

- 1) разработаны различные варианты реализации каждого этапа;
- 2) существенно изменена структура алгоритма.

Среди первой группы можно выделить различные методы селекции родителей (пропорциональный, турнирный, локальный, ранжирование и т. д.), сокращения популяции (элитарный, случайный, пропорциональный, селекционный и т. д.). Разработаны различные генетические операторы кроссинговера двоичных (одноточечный, многоточечный, однородный) и действительных значений (дискретная, промежуточная, линейная рекомбинация), мутации (однородная и неоднородная для двоичных и вещественных значений).

Среди второй группы можно отметить ГА с динамически изменяемой мощностью популяции (различные стратегии выбора срока жизни особи), адаптивные ГА с подстраиваемыми параметрами (вероятности кроссинговера и мутации), параллельные алгоритмы (модель островов, глобальная модель рабочих – хозяин, диффузионная модель), иерархические (многоуровневые) ГА.

2.2. Варианты реализации основных этапов классического ГА

2.2.1. Инициализация, или формирование исходной популяции

В настоящее время наиболее известными и применяемыми на практике являются 3 способа создания исходной популяции (начального множества решений).

Стратегия **«одеяла»** – формирование полной популяции, содержащей все возможные решения. Например, для n -разрядной хромосомы мы имеем всего 2^n вариантов решений, которые составляют полную популяцию.

Стратегия **«дробовика»** – генерация достаточно большого случайного подмножества решений.

Стратегия **«фокусировки»** – генерация множества решений, включающих разновидности одного решения.

Первый подход (стратегия **«одеяла»**) практически не реализуем даже для задач средней размерности, вследствие больших вычислительных затрат. Заметим, что при этом подходе начальная популяция не может развиваться, т. к. в ней уже содержатся все возможные решения. Третий способ используется тогда, когда есть предположение, что некоторое решение является вариацией известного

значения. В этом случае время поиска оптимального решения существенно сокращается, т. к. алгоритм начинает работу в окрестности оптимума. Чаще всего применяют второй способ. В этом случае в результате эволюции есть возможность перейти в другие подобласти поиска и каждая из них имеет сравнительно небольшое пространство поиска. В целом эффективность ГА, качество решения и дальнейшая эволюция в значительной степени определяются структурой и качеством начальной популяции. На практике часто применяют комбинацию второго и третьего способов. Сначала определяются особи с высокими значениями целевой функции, а затем случайно формируются начальные решения в этих подобластях.

2.2.2. Селекция хромосом (отбор)

При отборе конкурентоспособных особей используют функцию приспособленности (*fitness function*), как единственно доступный источник информации о качестве решения. Но различные методы отбора родителей по-разному используют эту информацию. Слишком сильное предпочтение лучшим особям ведет к чрезмерному сужению области поиска решения и чревато преждевременной сходимость в локальных экстремумах, и, наоборот, пренебрежение ими может привести к другой крайности – случайному поиску.

В классическом ГА вероятность каждой особи попасть в промежуточную популяцию пропорциональна ее приспособленности, т. е. работает *пропорциональный отбор* (*proportional selection*). Существует несколько способов реализации данного способа селекции хромосом:

– *stochastic sampling*. Пусть особи располагаются на колесе рулетки так, что размер сектора каждой особи пропорционален ее приспособленности. N раз запуская рулетку, выбираем требуемое количество особей для записи в промежуточную популяцию.

– *Remainder stochastic sampling*. Для каждой особи вычисляется отношение ее приспособленности к средней приспособленности популяции. Целая часть этого отношения указывает, сколько раз нужно записать особь в промежуточную популяцию, а дробная показывает её вероятность попасть туда ещё раз. Реализовать такой способ отбора удобно следующим образом: расположим особи на рулетке так же, как было описано. Теперь пусть у рулетки не одна

стрелка, а N , причем они отсекают одинаковые сектора. Тогда один запуск рулетки выберет сразу все N особей, которые нужно записать в промежуточную популяцию. Такой способ иллюстрируется на рис. 2.2.

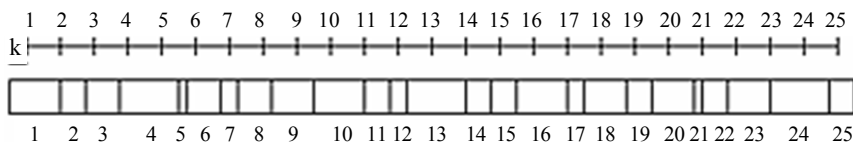


Рис. 2.2. Иллюстрация метода *remainder stochastic sampling*

Существуют также методы отбора, основанные не на вероятностном подходе.

Локальный отбор производится среди особей, которые находятся в некоторой ограниченной среде, где определено отношение соседства на различных структурах: линейное, двумерное (4-, 6-, 8-связное). При отборе родителей на первом шаге производится отбор особей случайным образом или одним из ранее рассмотренных способов. Далее для каждой отобранной особи определяется множество локальных соседей и среди них выбирается партнер для выполнения операций скрещивания. Предыдущие методы отбора в какой-то степени имеют аналогию с эволюцией естественных популяций. **Отбор на основе усечения** не имеет аналогов в естественной эволюции и обычно используются для больших популяций ($N > 100$). При этом сначала отбираемые особи упорядочиваются согласно их значениям целевой функции. Затем в качестве родителей выбираются только лучшие особи. Далее с равной вероятностью, среди них случайно выбирают пары, которые производят потомков. При этом методе используется параметр – порог отсечения (иногда используется термин интенсивность отбора), показывающий долю (часть популяции), которая отбирается в качестве родителей. Обычно используется порог от 10 % до 50 %. При **турнирном отборе** из популяции случайно отбираются m особей, затем лучшая из них выбирается в качестве родителя. Далее этот процесс повторяется, пока не сформируется промежуточная популяция, в которой случайно формируются пары родителей. Параметром этой процедуры является размер тура m , который принимает значения из диапазона от 2 до N .

2.2.3. Операторы кроссинговера

Различают операторы двоичной и вещественной рекомбинации (скрещивания).

Двоичный кроссинговер. В классическом ГА применяется *одноточечный* оператор кроссинговера (*1-point crossover*): для родительских строк случайным образом (чаще всего с вероятностью $p_c \approx 0,5$) выбирается точка раздела и производится обмен фрагментами хромосом после точки скрещивания, как показано в примере на рис. 2.3.

A:	0	1	1	1	0	1	0	1	0
B:	1	0	0	1	1	0	1	0	1
⇕									
A':	0	1	1	1	0	0	1	0	1
B':	1	0	0	1	1	1	0	1	0

Рис. 2.3. Одноточечный двоичный кроссинговер

В случае *многоточечного* кроссинговера выбираются m позиций $k_i \in 1; 2; \dots; N - 1$ и их номера сортируются по порядку. Затем родители обмениваются фрагментами, заключенными между соседними выбранными позициями и производят, таким образом, двух потомков. То есть обмен здесь идет секциями. При этом секции между первой позицией и первой точкой скрещивания не обмениваются, а далее обмен идет через одну секцию, как это показано на рис. 2.4.

A	0	1	1	1	0	0	1	1	0	1	0
B	1	0	1	0	1	1	0	0	1	0	1
			⇕								⇕
A':	0	1	1	0	1	1	1	1	0	1	1
B':	1	0	1	1	0	0	0	0	1	0	0

Рис. 2.4. Многоточечный двоичный кроссинговер

Для этого случайным образом генерируется двоичная маска кроссинговера той же длины (с тем числом бит), что и у хромосом родителей. Четность бита маски показывает родителя, из которого копируется ген потомка (например, 1 соответствует первому родителю, а 0 – второму). На рис. 2.5 показана схема выполнения этого типа кроссинговера на примере.

Маска ОК	1	0	0	1	0	1	1	1	0	0
1-й родитель	1	0	1	0	0	0	1	1	1	0
	↓			↓		↓	↓	↓		
Потомок	1	1	0	0	0	0	1	1	1	1
		↑	↑		↑				↑	↑
2-й родитель	0	1	0	1	0	1	0	0	1	1

Рис. 2.5. Однородный двоичный кроссинговер

При *ограниченном* кроссинговере точки скрещивания могут вбираться только там, где значения генов у родителей различны.

Кроссинговер действительных значений. Данная группа операторов разработана для хромосом, представленных действительными числами.

Дискретный кроссинговер определен над векторами, компонентами которых являются вещественные числа, и выполняется аналогично однородному кроссинговеру, определенному над двоичными векторами (рис. 2.6).

1-й родитель	12	25	5	17	39
2-й родитель	123	4	34	-5	11
1-й образец (маска)	2	2	1	1	2
1-й образец (маска)	1	2	1	2	1
1-й потомок	123	4	5	17	11
2-й потомок	12	4	5	-5	39

Рис. 2.6. Дискретный кроссинговер вещественных векторов

Промежуточный кроссинговер применим только для особей, представленных только вещественными значениями. Здесь значения потомков строятся в окрестности или между значениями родителей (рис. 2.7).

1-й родитель	12	25	5	17	39
2-й родитель	123	4	34	-5	11
Случайно выбраны следующие значения коэффициента a					
1-й образец (маска)	0,5	1,1	0,1	0,7	0,4
1-й образец (маска)	0,1	0,8	0,5	1,1	0,3
1-й потомок	67,5	1,9	2,1	1,6	27,8
2-й потомок	23,1	8,2	19,5	-7,2	30,6

Рис. 2.7. Промежуточный кроссинговер

Потомок формируется следующим образом:

$$A' = A + \lambda (B - A),$$

где A, B – вещественные значения – родители;

A' – вещественное значение – потомок;

$\lambda \in [-d; d + 1]$ – случайный масштабирующий множитель.

В случае обычного промежуточного кроссинговера $d = 0$ и $\lambda \in [0; 1]$. Для обобщенного промежуточного кроссинговера $d > 0$. Значение каждого потомка формируется по приведенному выражению, но со своим λ . В случае особи-вектора случайным образом формируется вектор масштабирующий множители $\lambda = (\lambda_1; \lambda_2; \dots; \lambda_n)$ и потомок-вектор определяется покомпонентно. Фактически, этот оператор заимствован из другого направления эволюционных вычислений – «эволюционные стратегии».

Линейный кроссинговер аналогичен предыдущему, за исключением того, что значение масштабирующего множителя a одинаково для всех компонент векторов.

2.2.4. Операторы мутации

К полученному в результате отбора и скрещивания новому поколению (потомкам) применяется оператор мутации, необходимый

для «выбивания» популяции из локального экстремума и способствующий защите от преждевременной сходимости. Мутация может быть выполнена различными способами.

Двоичная мутация. В классической мутации каждый бит каждой особи популяции с некоторой вероятностью p_m инвертируется. Эта вероятность обычно очень мала, менее 1 %.

Иногда используется оператор инверсии, фактически являющийся разновидностью классической мутации. При этом случайным образом выбираются две позиции в особи, и далее производится обмен значениями генов между ними. Пример выполнения этого оператора представлен на рис. 2.8.

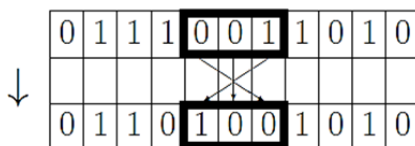


Рис. 2.8. Мутация-инверсия

Можно выбирать некоторое количество точек в хромосоме для инверсии, причем их число также может быть случайным. Среди рекомендаций по выбору вероятности мутации нередко можно встретить варианты $1/L$ или $1/N$.

Мутация над вещественными числами. Мутация над вещественными потомками выполняется путем сложения особи с небольшим случайным значением, которое называется шагом мутации. Выбор размера шага мутации зависит от рассматриваемой проблемы, и шаг в общем случае может изменяться в процессе решения задачи. Маленький шаг дает большую точность, но ведет к большим временным затратам.

2.2.5. Стратегии формирования нового поколения

Можно выделить два основных типа формирования нового поколения после кроссовера и мутации:

- дети замещают родителей;
- новое поколение составляется из совокупности детей и их родителей.

Также применяется принцип *элитизма*: в новое поколение включается заданное количество лучших особей предыдущего поколения (часто одна лучшая особь).

Использование второй стратегии и элитизма не допускает потери лучших решений. К примеру, если популяция сошлась в локальном максимуме, а мутация вывела одну из строк в область глобального, то при замещении родителей весьма вероятно, что эта особь в результате скрещивания будет потеряна, и решение задачи не будет получено. Если же используется элитизм, то полученное хорошее решение будет оставаться в популяции до тех пор, пока не будет найдено лучшее.

2.2. ГА с изменяемой мощностью популяции

В этой модификации ГА отсутствуют этап и механизм селекции родителей. Вместо этого вводится концепция возраста особи. Каждой особи после ее рождения на текущем этапе оценки функции приспособленности присваивается *life time* (срок жизни) – параметр, зависящий от функции приспособленности особи. Таким образом, каждая особь живет определенное число поколений и умирает по окончанию заданного срока жизни. Очевидно, значение параметра *life time* влияет на размер популяции. Чем больше срок жизни, тем в большем числе поколений популяция живет данная особь, что увеличивает средний размер популяции.

Структура такого ГА выглядит следующим образом:

1. Инициализация или формирование исходной популяции.
2. Вычисление функции приспособленности (вычисление пригодности) хромосом в популяции.
3. Определение срока жизни хромосом.
4. Вычисление условия окончания процесса.
5. Увеличение возраста каждой хромосомы на 1.
6. Селекция хромосом с целью формирования пар родителей.
7. Применение генетического оператора скрещивания (*recombination*) хромосом промежуточной популяции путем кроссинговера или кроссовера (*crossover*), что приводит к формированию нового поколения.
8. Применение генетического оператора мутации (*mutation*).
9. Определение срока жизни хромосом.

10. Удаление из популяции всех хромосом с возрастом больше срока жизни.

11. Формирование результата – выбор наилучшей хромосомы.

2.3. Последовательные модификации классического ГА

2.3.1. Алгоритм *Genitor*

В данной модели используется специфичная стратегия отбора. На каждом шаге только *одна* пара случайных родителей создает только *одного* ребенка. Этот ребенок заменяет не родителя, а одну из худших особей популяции.

Таким образом, на каждом шаге в популяции обновляется *только* одна особь.

2.3.2. Алгоритм *SteadyState (SSGA)*

Алгоритм SSGA напоминает предыдущий. SSGA не формирует промежуточную популяцию как классический ГА, а осуществляет последовательно выбор пары наилучших индивидов. Затем к этой паре применяются генетические операторы скрещивания и мутации с целью формирования наследников, которые заменяют худших индивидов популяции.

2.3.3. ГА селекционеров

ГА селекционеров использует знания, полученные селекционерами в области искусственной селекции. В этом алгоритме «виртуальный селекционер» отбирает некоторый процент хромосом из первоначальной популяции для создания потомков. Именно этим алгоритм отличается от классического ГА. Все остальные этапы остаются неизменными.

2.3.4 Алгоритм *CHC*

CHC – это *Cross generational elitist selection, Heterogenous recombination, Cataclysmic mutation*.

В этом алгоритме для нового поколения выбираются *N* лучших *различных* особей среди родителей и детей. Дублирование строк не допускается. Для скрещивания все особи разбиваются на пары, но

скрещиваются только те пары, между которыми расстояние Хэминга больше некоторого порогового (также возможны ограничения на минимальное расстояние между крайними различающимися битами). При скрещивании используется так называемый HUX-оператор (Half Uniform Crossover), разновидность однородного кроссовера – каждому потомку переходит ровно половина битов каждого родителя. Размер популяции небольшой. Этим оправдано использование однородного кроссовера.

В СНС применяется *cataclysmic mutation*: все строки, кроме самой приспособленной, подвергаются сильной мутации (изменяется около трети битов). Таким образом, алгоритм перезапускается и далее продолжает работу, применяя только кроссовер.

2.3.5. Гибридный ГА

Использование гибридного алгоритма позволяет объединить преимущества ГА с преимуществами классических методов.

Дело в том, что ГА являются робастными алгоритмами, т. е. позволяют находить хорошее решение, но нахождение оптимального решения зачастую оказывается намного более трудной задачей в силу стохастичности принципов работы алгоритма. Поэтому возникла идея использовать ГА на начальном этапе для эффективного сужения пространства, поиска вокруг глобального экстремума, а затем, взяв лучшую особь, применить один из «классических» методов оптимизации.

Однако можно использовать «классические» методы (*hill-climbing*, например) и внутри самих ГА. На каждом поколении каждый полученный потомок оптимизируется этим методом, таким образом, каждая особь достигает локального максимума, вблизи которого она находится, после чего подвергается отбору, скрещиванию и мутации. Такой метод ухудшает способность алгоритма искать решение с помощью отбора гиперплоскостей, но зато возрастает вероятность того, что одна из особей попадет в область глобального максимума и после оптимизации окажется решением задачи.

2.4. Параллельные модификации классического ГА

ГА применяются и при параллельных вычислениях. При этом формируется несколько живущих отдельно популяций. На этапе

формирования нового поколения по некоторому правилу отбираются особи из разных популяций. Сгенерированная таким образом популяция в некоторых случаях заменяет все остальные. В результате, так или иначе, происходит миграция особей из одной популяции в другие. Поэтому параллельные ГА иногда называют миграциями.

2.4.1. Параллельный ГА на базе популяций

В этом алгоритме используются $N/2$ процессов при популяции в N индивидов (здесь и далее процесс рассматривается как некоторая машина, процессор, который может работать независимо). Каждый из них дважды случайным образом выбирает из общей памяти две особи и оставляет лучшую. Две выбранные особи затем подвергаются скрещиванию, мутации и оценке степени пригодности. Полученные потомки размещаются в общей памяти.

2.4.2. Миграция

Миграционная модель (модель островов) разбивает популяцию решений на множество подпопуляций (одна большая разбивается на множество мелких). Эти подпопуляции функционируют независимо друг от друга некоторое число поколений (срок изоляции). По достижению срока изоляции производится обмен между подпопуляциями. При этом задаются:

- число обмениваемых особей – степень миграции;
- метод отбора особей для миграции;
- схема миграции, которая определяет способ обмена между подпопуляциями.

Выбор особей для миграции может осуществляться несколькими методами, как и отбор родителей. Существует также несколько стандартных схем обмена отобранными особями между подпопуляциями:

- каждая подпопуляция обменивается с каждой из остальных подпопуляций;
- обмен по кольцу;
- обмен между соседними подпопуляциями.

Параллельная реализация миграционной модели показала не только увеличение скорости вычислений, но и меньшее число событий – вычислений значений функции приспособленности для особей по

сравнению с классическим ГА (одна популяция). Поэтому даже для однопроцессорного компьютера последовательная реализация параллельного алгоритма (псевдопараллельная реализация) обычно дает лучшие результаты.

2.4.3. Модель «Рабочий – Хозяин»

В глобальной модели «Рабочий – Хозяин» на нижнем уровне каждый «рабочий» имеет свою популяцию, каждая из которых развивается самостоятельно. На верхнем уровне «Хозяин» получает с нижнего уровня значения функции приспособленности из всех подпопуляций и управляет ими путем коррекции основных параметров ГА нижнего уровня таких как мощность популяции, вероятности скрещивания и мутации и т. п.

2.4.4. Диффузионная модель

Диффузионная модель работает с каждой особью индивидуально и выбирает партнера для скрещивания подобно локальному отбору родителей. Таким образом, имеет место «диффузия» информации в популяции.

3. ОСНОВНЫЕ ЭТАПЫ И ТРЕБОВАНИЯ К ВЫПОЛНЯЕМОЙ РАБОТЕ

Цель работы: В ходе выполнения лабораторной работы необходимо разработать программный модуль, позволяющий сформировать оптимальную структуру заданного ГА. Рассмотрим подробнее требования к функциональности программного модуля.

Тема: Реализация распределенного параллельного генетического алгоритма (РПГА).

Начальные условия: наброс начальных точек – треугольная сетка (несколько заданных заранее вариантов); селекция – турнир; скрещивание – двухточечное; мутация – задается вероятностью $[0; 1]$; параметры X_1, X_2 изменяются в интервале $[1; 5]$ с точностью 0,1. Функция $Y = \ln(X_1) + X_2$ минимизируется.

Описание алгоритма: Используются независимые конкурирующие подпопуляции, которые обмениваются особями с заданной частотой. Шаги:

0. Инициализация. Определение генетического представления задачи.

1. Создание первоначальной популяции и разделение ее на подпопуляции.

2. Для каждой подпопуляции выполняются параллельно следующие шаги:

2.1. Применение в течение заданных поколений селекции, скрещивания и мутации.

2.2. Перемещение нескольких хромосом в соседние подпопуляции.

2.3. Получение такого же количества хромосом из соседних подпопуляций.

3. Возврат к шагу 2.

Реализация: Программа представляет собой приложение WindowsForms, написанное на языке C#. Интерфейс представлен на рис. 3.1.

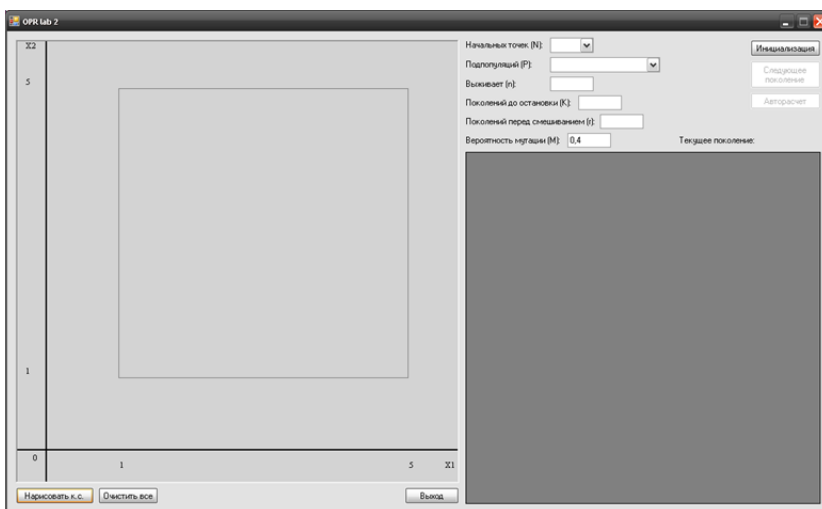


Рис. 3.1. Главное окно программы

Параметры, задаваемые пользователем:

- Количество начальных точек (b , δ , 24).
- Количество подпопуляций (P).
- Выживает особей, т. е. переходит на следующее поколение (n).
- Поколений до остановки (K).

- Поколений перед смешиванием (r).
- Вероятность мутации (M).

После задания параметров нажимаем кнопку «Инициализация». Происходит генерация начальной популяции (поколение под номером 0), разбиение особей на подпопуляции (каждая подпопуляция имеет номер, начиная с 1). Каждая точка в подпопуляции имеет свой номер, начиная с 1. Идентификатор точки составляется из номера подпопуляции и номера точки (например, «1.2»).

Можно произвести автоматический расчет, нажав кнопку «Авторасчет» (рис. 3.2). Шаги генетического алгоритма выполняются для заданного количества поколений, и лучшая особь выводится отдельной строкой в конце таблицы, а также отмечается жирной зеленой точкой на графике.

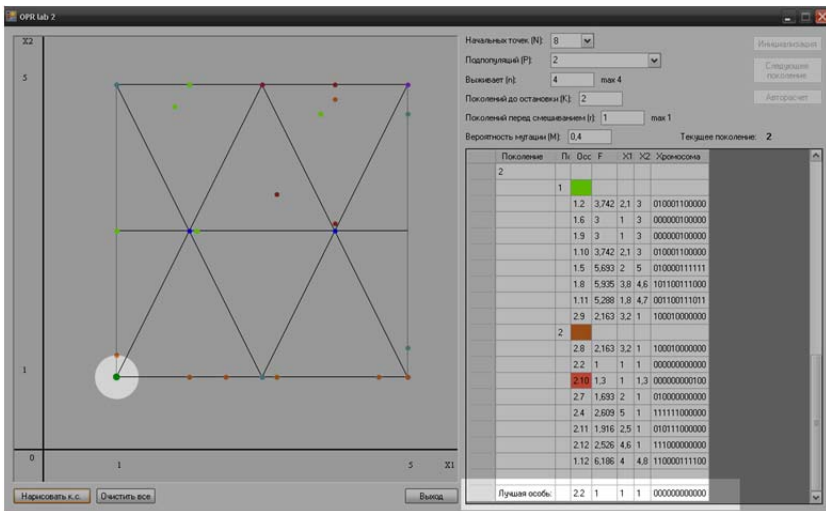


Рис. 3.2. Результат авторасчета

Переходить на каждое следующее поколение можно нажатием кнопки «Следующее поколение» (рис. 3.3). В этом случае новое поколение добавляется в конец таблицы, а особи подпопуляций отмечаются на графике точками (каждая подпопуляция имеет свой цвет).

Таблица содержит информацию о каждой особи: значение приспособленности (функции) F ; значения параметров $X1$ и $X2$; значения хромосомы (параметров $X1$ и $X2$, переведенные в двоичный код).

При нажатии кнопки «Следующее поколение» происходят:

1. *Селекция.* В каждой подпопуляции особи разбиваются на пары случайным образом и из каждой пары выбирается лучшая (по значению приспособленности). Она выживает. Из каждой подпопуляции выбирается указанное пользователем количество особей.

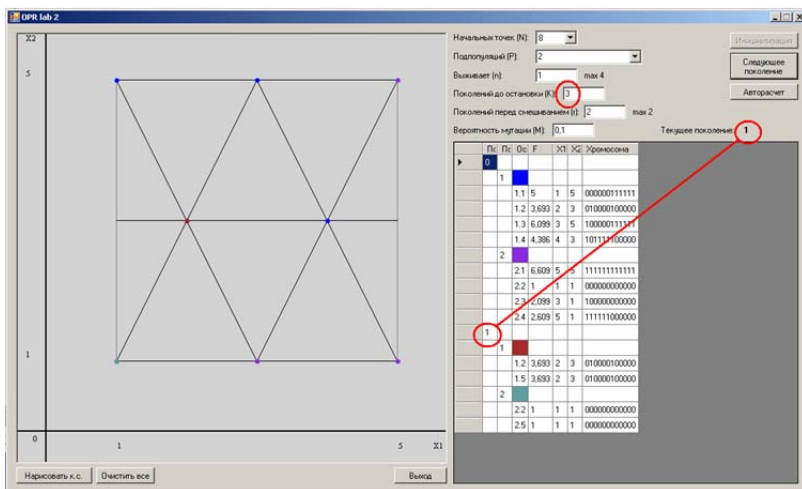


Рис. 3.3. Первое поколение из трех

2. *Скрещивание.* В каждой подпопуляции особи разбиваются на пары случайным образом. Каждая пара дает два потомка в соответствии с правилами двухточечного скрещивания (точки выбираются случайным образом). Если в подпопуляции нечетное количество особей, то оставшаяся без пары особь дает одного потомка инверсией (две точки для инверсии выбираются случайным образом).

3. *Мутация.* Для каждой особи генерируется случайное число из интервала $[0; 1]$. Если оно меньше указанной пользователем вероятности мутации, то особь мутирует. Новая особь при этом **не рождается**. Мутировавшая особь отмечается в таблице более темным цветом (рис. 3.4).

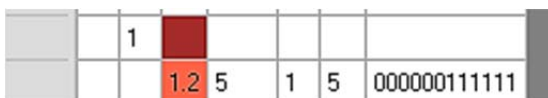


Рис. 3.4. Особь 1.2 мутировала

Мутации подвергаются как выжившие особи (с прошлого поколения), так и их потомки, образованные на шаге 2.

4. По прошествии указанного пользователем количества поколений происходит *смешивание подпопуляций*. Подпопуляции случайным образом разбиваются на пары. Внутри каждой пары первая подпопуляция «отдает» второй одну особь, а вторая отдает одну особь первой. При нечетном количестве подпопуляций одна подпопуляция остается без пары, ее особи никуда не переходят (рис. 3.5).

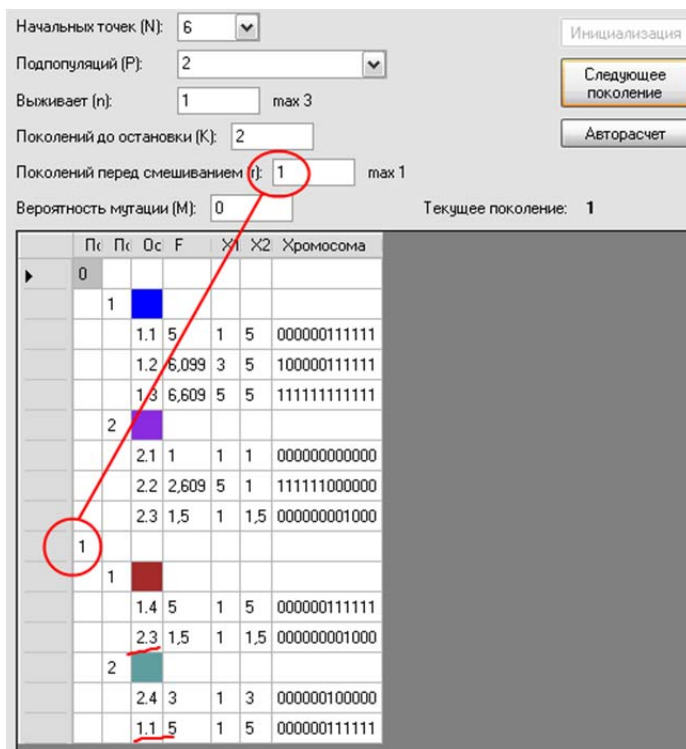


Рис. 3.5. Смешивание подпопуляций

На рис. 3.5 видно, что особь 1.1 в результате смешивания перешла в 2, а особь 2.3 перешла в 1.

После выполнения заданного количества шагов (поколений) находится лучшая особь и выводится отдельной строкой в конце таблицы, а также отмечается жирной зелёной точкой на графике.

Библиографический список

1. Большаков, В. Твердотельное моделирование деталей в САД-системах : AutoCAD, КОМПАС-3D, SolidWorks, Inventor, Creo : 3D-модели и конструкторская документация сборок / В. Большаков [и др.]. – Санкт-Петербург : Питер, 2015. – 473 с.
2. Большаков, В. П. Основы 3D-моделирования. Изучаем работу в AutoCAD, КОМПАС-3D, SolidWorks, Inventor : учебное пособие для студентов вузов по направлению 211000 «Конструирование и технологии электронных средств» / В. П. Большаков и А. Л. Бочков. – Санкт-Петербург : Питер, 2013. – 300 с.
3. Тику, Шам. SolidWorks 2005 / Шам Тику [и др.] ; пер. А. Вахитов. – Санкт-Петербург : Питер, 2006. – 815 с.
4. Пашкевич, М. Ф. Технологическая оснастка : учебник для студентов машиностроит. Специальностей вузов / М. Ф. Пашкевич [и др.]. – Минск : Адукацыя і выжаванне. 2002. – 320 с.
5. Гарнаев, А. Ю. Visual Basic.NET : разработка приложений / А. Ю. Гарнаев. – Санкт-Петербург : БХВ-Петербург, 2002. – 624 с. : ил.
6. АвтоСпецТехника – машины для работы. – Режим доступа : <http://asptech.ru>. – Дата доступа : 20.09.2016.
7. КБ-403 – самомонтирующийся грузоподъемный агрегат // AllSpecTech.com. – Режим доступа : <http://allspectech.com/stroitelnyaya/kranu/bashennyye/kb-403.html>. – Дата доступа : 28.09.2016.
8. Башенные краны. – Режим доступа : <http://stroy-technics.ru/bashennyye-kranu/> – Дата доступа : 10.08.2016.
9. Савицкий, Г. А. Ветровая нагрузка на сооружения / Г. А. Савицкий. – Москва, 1972.
10. Нагрузки и воздействия. Нормы проектирования : СНиП 2.10.07-85.
11. Бобров, А. В. Ветро дизельные комплексы в децентрализованном электроснабжении : монография / А. В. Бобров, В. А. Тремясов. – СФУ. – 2012. – 216 с.
12. Справочная система ANSYS HELP.
13. Holland, J. P. Adaption in natural and artificial systems. Anarbor : the uni of Michigan press, 1975.
14. Goldberg, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning / D. E. Goldberg. – Addison-Wesley Publishing Company, 1989. – 412.

15. Michalewicz, Z. Genetic Algorithms + Data Structures = Evolution Programs / Z. Michalewicz. – Springer-Verlag, 1994. – 340.
16. Mitchell, M. An Introduction to Genetic Algorithms / M. Mitchell. – MIT Press, 1998. – 209.
17. Genetic and evolutionary Algorithms : Principles, Methods and Algorithms. GEA Toolbox.
18. Курейчик, В. М. Генетические алгоритмы / В. М. Курейчик. – Таганрог : Изд-во ТРТУ, 1998.
19. Herrera, F. Adaptive genetic operators based on coevolution with fuzzy behaviors / Herrera F., Lozano M. // IEEE transactions on evolutionary computation. – 2001. – Vol. 5, № 2. – P. 149–165.

Учебное издание

БОРОДУЛЯ Алексей Валентинович
КОВАЛЁВА Ирина Львовна
КОЧУРОВ Вадим Александрович и др.

**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
В АВТОМАТИЗИРОВАННОМ ПРОЕКТИРОВАНИИ
МАШИНОСТРОИТЕЛЬНЫХ КОНСТРУКЦИЙ**

Учебно-методическое пособие
для студентов специальности 1-40 05 01-01
«Информационные системы технологии
(в проектировании и производстве)»

В 3 частях

Часть 1

Редактор *А. С. Кириллова*
Компьютерная верстка *Н. А. Школьниковой*

Подписано в печать 22.07.2020. Формат 60×84 ¹/₁₆. Бумага офсетная. Ризография.
Усл. печ. л. 5,74. Уч.-изд. л. 4,50. Тираж 100. Заказ 1047.

Издатель и полиграфическое исполнение: Белорусский национальный технический университет.
Свидетельство о государственной регистрации издателя, изготовителя, распространителя
печатных изданий № 1/173 от 12.02.2014. Пр. Независимости, 65. 220013, г. Минск.