



МИНИСТЕРСТВО ОБРАЗОВАНИЯ
РЕСПУБЛИКИ БЕЛАРУСЬ

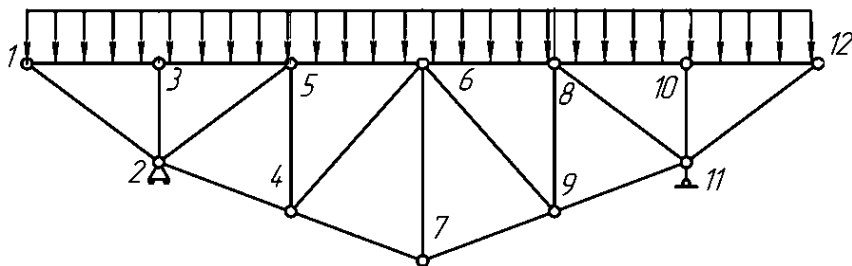
Белорусский национальный
технический университет

Кафедра «Программное обеспечение
информационных систем и технологий»

КОМПЬЮТЕРНОЕ КОНЕЧНО-ЭЛЕМЕНТНОЕ МОДЕЛИРОВАНИЕ

Пособие

Часть 1



Минск
БНТУ
2021

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Белорусский национальный технический университет

Кафедра «Программное обеспечение
информационных систем и технологий»

КОМПЬЮТЕРНОЕ КОНЕЧНО-ЭЛЕМЕНТНОЕ МОДЕЛИРОВАНИЕ

Пособие для студентов специальности
1-40 05 01 «Информационные системы и технологии
(по направлениям)» направления специальности
1-40 05 01-01 «Информационные системы и технологии
(в проектировании и производстве)»

В 2 частях

Часть 1

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

Минск
БНТУ
2021

УДК 004.9+519.6(075.8)

ББК 22.19я7

К64

А в т о р ы:

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

Р е ц е н з е н т ы:

заведующий лабораторией «Синтез технических систем»
ОИПИ НАН Беларуси, д-р техн. наук *С. В. Медведев*;
кафедра «Полиграфическое оборудование и системы
обработки информации» БГТУ, зав. кафедрой,
канд. техн. наук, доцент *М. С. Шмаков*;
доцент кафедры, канд. техн. наук *В. П. Беляев*

К64

Компьютерное конечно-элементное моделирование : пособие для студентов специальности 1-40 05 01 «Информационные системы и технологии (по направлениям)» направления специальности 1-40 05 01-01 «Информационные системы и технологии (в проектировании и производстве)» : в 2 ч. / В. В. Напрасников [и др.]. – Минск : БНТУ, 2021. – Ч. 1. – 83 с.

ISBN 978-985-583-273-8 (Ч. 1).

Настоящий материал предназначен для использования в качестве методических указаний при выполнении лабораторных, курсовых и дипломных работ студентами специальности 1-40 05 01 «Информационные системы и технологии», а также будет полезен магистрантам и аспирантам других технических специальностей 1-40 80 04 «Математическое моделирование, численные методы и комплексы программ», 1-31 81 12 «Прикладной компьютерный анализ данных», научные интересы которых связаны с конечно-элементным моделированием.

УДК 004.9+519.6(075.8)

ББК 22.19я7

ISBN 978-985-583-273-8 (Ч. 1)

ISBN 978-985-583-274-5

© Белорусский национальный
технический университет, 2021

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1. ОБЩИЕ СВЕДЕНИЯ	6
1.1. Организация комплекса	6
1.2. Модули комплекса	6
1.3. Знакомство с интерфейсом системы ANSYS	7
2. АНАЛИЗ ЗАДАЧИ В ANSYS	11
2.1. Построение модели	11
2.1.1. Присвоение имени файлу базы данных	11
2.1.2. Определение заголовка	11
2.1.3. Определение единиц измерения	12
2.1.4. Определение типа элемента	12
2.1.5. Определение опций элемента	13
2.1.6. Определение констант элемента	14
2.1.7. Определение свойств материала	14
2.1.8. Создание конечно-элементной модели	15
2.1.9. Приложение нагрузок	15
2.2. Решение задачи	15
2.2.1. Определение типа анализа	15
2.2.2. Спецификация решения	16
2.2.3. Решение задачи	17
2.3. Обзор результатов решения задачи	19
2.4. Общая структура командного файла ANSYS	19
3. ГЕОМЕТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ В ANSYS	21
3.1. Моделирование «снизу вверх»	21
3.2. Моделирование «сверху вниз»	28
4. РАСЧЕТ ПЛОСКОЙ ФЕРМЫ В ANSYS	33
4.1. Какая конструкция называется фермой	33
4.2. Пример расчета фермы с использованием графического интерфейса пользователя	34
4.2.1. Построение геометрической части модели	36
4.2.2. Выбор типа конечного элемента	39
4.2.3. Задание свойств материала	40
4.2.4. Построение конечно-элементной сетки	40
4.2.5. Задание условий закрепления конструкции	41
4.2.6. Задание внешних сил, приложенных к конструкции	43
4.2.7. Запуск решателя и просмотр результатов	45

4.2.8. Анализ результатов решения.....	51
5. АВТОМАТИЗАЦИЯ РАБОТЫ В ANSYS	52
5.1. Общие сведения о языке APDL.....	57
5.2. Переменные в APDL и ANSYS	58
5.3. Описание некоторых команд APDL	60
5.4. Командный файл для решения задачи о ферме.....	69
6. ВЫПОЛНЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ.....	72
6.1. Оформление отчета	72
6.2. Задания к расчету	73
7. КОНТРОЛЬНЫЕ ВОПРОСЫ.....	80
ЛИТЕРАТУРА	81

ВВЕДЕНИЕ

Широкое распространение современной вычислительной техники за последние 10–15 лет существенно изменило процесс инженерной деятельности. Появление на рынке программного обеспечения современных комплексов САД и САЕ позволяет ускорять процессы проектирования и исследования различных конструкций, в том числе и машиностроительных. Использование современных методов вычислений, реализованных в комплексах САД, дает возможность проводить исследования различных характеристик проектируемых объектов, что позволяет менять конструкцию этих объектов без создания экспериментальных образцов и не прибегать к дорогостоящим натурным испытаниям.

Среди средств САЕ (средств обеспечения исследований) важное место занимают комплексы метода конечных элементов (МКЭ, FEA), которые позволяют проводить имитационное моделирование работы исследуемой конструкции на основе подробного описания ее геометрии, физики моделируемых процессов, свойств применяемых материалов, эксплуатационных характеристик и иных указываемых пользователем исходных и начальных данных. Среди комплексов МКЭ можно указать такие продукты, как ANSYS, COSMOS, MSC/NASTRAN и другие.

Комплекс МКЭ ANSYS позволяет инженерам-исследователям проводить исследования не только характеристик динамики и прочности машиностроительных, строительных и иных конструкций (то есть расчеты задач механики деформируемого твердого тела, МДТТ), но и расчеты задач определения полей температур, динамики жидкости и газа, электромагнитных и акустических полей.

Комплекс МКЭ ANSYS применительно к прочности позволяет решать задачи статические (линейные, а также физически и геометрически нелинейные), определять собственные частоты модели (собственные колебания), исследовать поведение модели при воздействии гармонически изменяющихся нагрузок (вынужденные колебания), задачи линейной и нелинейной устойчивости, а также линейные и нелинейные динамические переходные процессы и т. д.

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Организация комплекса

Начальный уровень является уровнем входа в комплекс ANSYS или выхода из него. Он также используется для проведения ряда общих контрольных действий в комплексе, таких как изменение имени задания (jobname), очистки (обнуления базы данных), копирования двоичных (бинарных) файлов. При вызове комплекса открывается начальный уровень.

Уровень модуля является уровнем одной из доступных процедур. Каждый модуль является набором функций, выполняющих специфические действия при расчете. Например, общий препроцессор (PREP7) предназначен для создания модели, модуль расчета (SOLUTION) предназначен для приложения нагрузок и получения расчетных результатов, общий постпроцессор (POST1) предназначен для просмотра результатов решения. Дополнительный постпроцессор (POST26) предназначен для просмотра результатов в указанных пользователем точках модели в виде функции времени.

1.2. Модули комплекса

В общем случае вызов модуля проводится из главного (экранного) меню ANSYS *Main Menu* графического интерфейса пользователя (GUI). Например, вызов *Main Menu* ⇒ *Preprocessor* влечет вход в модуль PREP7. В качестве альтернативы для входа в модуль можно использовать команду (формат которой выглядит как /name, где name является именем модуля). Список модулей, их функций и команд, вызывающих вход в модуль, находится в табл. 1.1.

Таблица 1.1

Модули комплекса ANSYS

Модуль	Функция	Команда
PREP7	Создание модели (построение геометрии, указание материалов и т. д.)	/PREP7
SOLUTION	Приложение нагрузок и получение расчета МКЭ	/SOLU
POST1	Просмотр полученных результатов для указанного момента времени	/POST1

Модуль	Функция	Команда
POST26	Просмотр полученных результатов для указанных точек модели в виде функции времени	/POST26
OPT	Улучшение проекта (оптимизация)	/OPT
PDS	Определение влияния разброса и распределения исходных данных на результаты окончательного расчета	/PDS
AUX2	Перевод двоичных файлов в форму, пригодную для чтения	/AUX2
AUX12	Определение коэффициентов излучения и создание матриц излучения для теплового расчета	/AUX12
AUX15	Импорт файлов, созданных средствами CAD или другими комплексами МКЭ	/AUX15
RUNSTAT	Предварительное определение времени расчета, ширины матрицы жесткости и иных характеристик для расчета	/RUNST

1.3. Знакомство с интерфейсом системы ANSYS

Существует два режима работы в классическом варианте среды ANSYS: интерактивный и командный.

Интерактивный режим позволяет видеть результат каждой проведенной в ANSYS операции.

Командный режим позволяет передать командный файл или набор команд, которые будут обработаны ANSYS в фоновом режиме.

В пособии рассматривается интерактивный режим, работая в нем, пользователь должен выбрать некоторую последовательность команд для проведения той или иной операции. Для облегчения восприятия все такие цепочки команд будут обозначены в пособии через знак «>».

К сожалению, на данный момент не существует русифицированной версии программы, поэтому названия всех команд приводятся на английском языке.

Запуск программы (11 версия) производится через последовательность команд *Start > Programs > ANSYS 11.0 > ANSYS Product Launcher*.

Появляется окно *Product Launcher* (рис. 1.1).

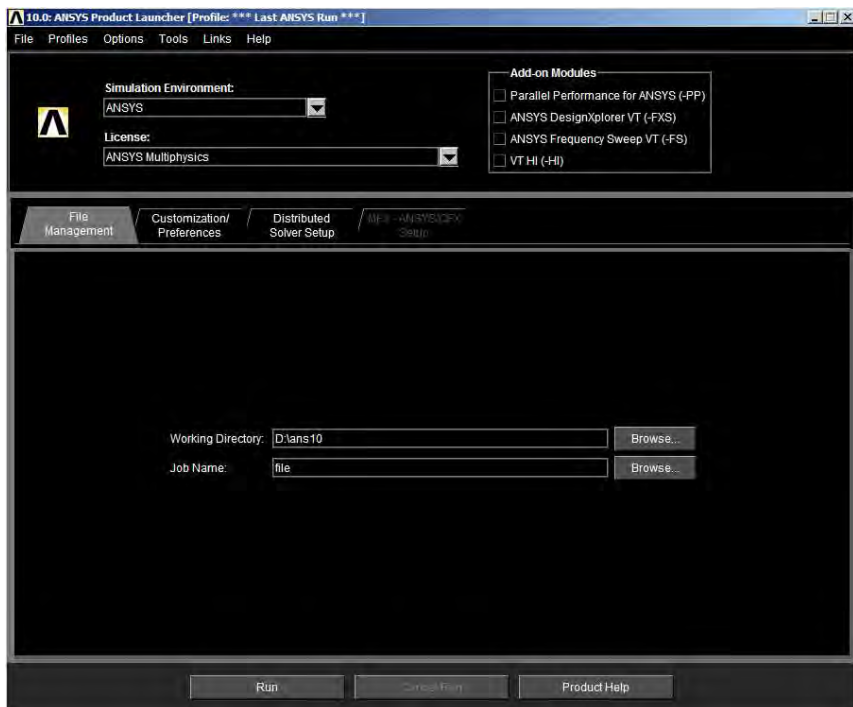


Рис. 1.1. Окно *Product Launcher*

В этом окне можно настроить рабочую директорию, название работы, использование памяти программой, работу в многопроцессорном режиме и др. Необходимо провести следующие настройки:

- *Simulation Environment* – выбор рабочего окружения (ANSYS для классического, интерактивный режим, ANSYS Batch – командный режим, ANSYS Workbench и др.);

- *Licenses* – выбор доступной лицензии;

- *Working directory* – рабочая директория, где хранятся по умолчанию все рабочие файлы. Эта директория должна быть создана заранее, кнопка справа позволяет производить выбор папок и локальных дисков;

- *Graphics device name* – определяет, какое устройство будет отвечать за прорисовку изображения: 3D видеокарта, Win32x процессор;

– *Job Name* – название текущей работы. Это название будет использовано при формировании файлов. Имя должно содержать только латинские буквы и цифры.

Удобно для каждой новой работы создавать отдельную директорию. Основными файлами являются:

– x.db – база расчета, где хранится описание модели, материалов, конечно-элементная сетка и т. д.;

– x.rst – файл результатов (результаты могут также находиться в предыдущем файле).

Working Directory и *Job Name* могут быть изменены в ходе выполнения работы.

После нажатия кнопки *Run* запускается интерфейс программы (рис. 1.2).

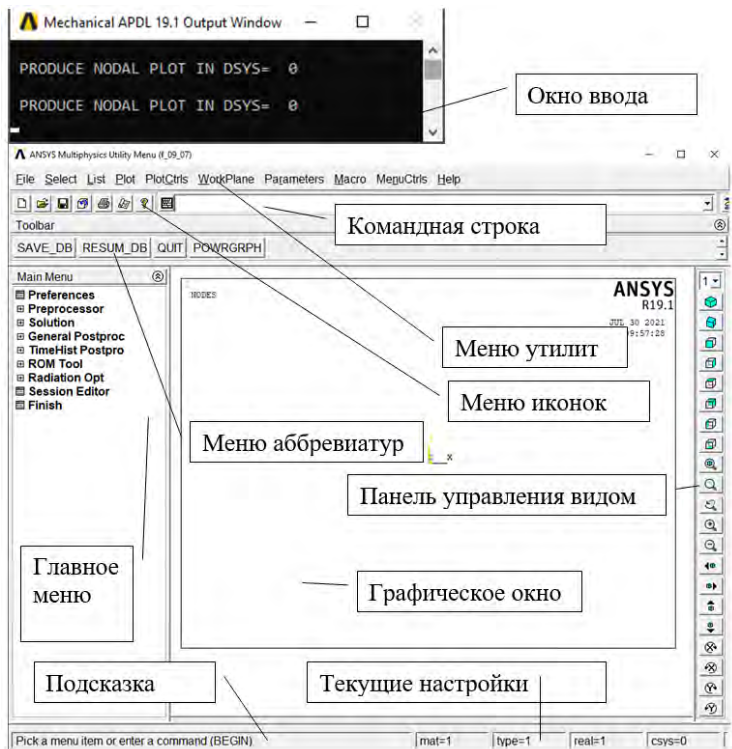


Рис. 1.2. Интерфейс программы

Можно выделить основные элементы интерфейса:

1. *Меню утилит* (управление файлами, выбором и компонентами, проекциями, параметрами):

1.1. *File* – работа с файлами;

1.2. *Select* – выбор объектов (по типу, мышкой, по ассоциативности и т. д.);

1.3. *List* – информация о модели в текстовом виде;

1.4. *Plot* – выбор объектов для рисования по типу;

1.5. *PlotCtrls* – настройка параметров отображения;

2. *Строка ввода команд* (командная строка). Используется для ввода команд;

3. *Главное меню* (основное рабочее меню препроцессора, решателя и постпроцессора). Представляет собой раскрывающийся вложенный список команд, сгруппированных по назначению;

4. *Графическое окно* (окно вывода графических изображений);

5. *Окно вывода*. Здесь выводится отклик программы на действия пользователя, в том числе комментарии к ошибкам. Это окно может перекрываться основным окном.

2. АНАЛИЗ ЗАДАЧИ В ANSYS

Анализ любой задачи в ANSYS состоит из следующих этапов:

- построение модели;
- решение задачи;
- постпроцессорная обработка результатов.

Для выполнения задач на каждом этапе используется свой процессор.

2.1. Построение модели

Моделирование объекта – это основной и самый трудоемкий этап решения задачи. Моделирование производится в препроцессоре PREP7. На этом этапе исходя из математических моделей механики задается геометрическая модель объекта, определяются типы используемых элементов, задаются свойства материала и краевые условия. Будем рассматривать поэтапно типичные действия, выполняемые при построении модели. Действия рассмотрим как через ГИП (графический интерфейс пользователя), так и через команды.

2.1.1. Присвоение имени файлу базы данных

При выполнении этой операции файлу базы данных и всем сопутствующим файлам присваивается новое имя. Если этого не выполнить, то файлы будут иметь имя по умолчанию file с соответствующим расширением. Для присвоения имени необходимо выполнить путь в меню ГИП и ввести свое имя файла:

Utility Menu > File > Change Jobname

либо воспользоваться соответствующей командой в строке ввода:

/FILENAME,filename.

2.1.2. Определение заголовка

Для определения заголовка необходимо выполнить путь в меню ГИП:

Utility Menu > File > Change Title.

При выполнении этой операции в графическом окне появится название заголовка. Для достижения той же цели можно воспользоваться соответствующей командой:

/TITLE,filename.

2.1.3. Определение единиц измерения

Поскольку в расчетах по умолчанию используется британская система мер, то для перехода к системе единиц СИ необходимо выполнить команду */UNITS*. Данная команда недоступна из ГИП и должна непосредственно вводиться в строку ввода (командную строку):

/UNITS,SI.

2.1.4. Определение типа элемента

Библиотека элементов ANSYS содержит более 100 различных типов конечных элементов. Каждый элемент имеет свое имя, описывающее семейство элементов, необходимых для моделирования соответствующего объекта, и номер. В табл. 2.1 приведены некоторые из них.

Таблица 2.1

Основные типы конечных элементов ANSYS

Тип элемента	Область использования
LINK	Моделирование ферменных конструкций, тросов, канатов и т. д.
BEAM	Моделирование стержневых конструкций
SHELL	Моделирование тонкостенных конструкций
PLANE	Моделирование двумерных задач (плоская задача, плоско-напряженное состояние, осесимметричная задача)
SOLID	Моделирование трехмерных объектов
PIPE	Моделирование стержневых систем «труба + жидкость»
MASS	Моделирование абсолютно твердого тела и материальной точки
CONTAC	Моделирование условий контакта
COMBIN	Моделирование пружин с различными свойствами (упругие, вязкоупругие и т. д.)

Типом элемента определяются:

– степени свободы элемента (которые, в свою очередь, влияют и на тип анализа – механический, термический, магнитный, электрический);

– модель объекта – одномерная, двумерная или трехмерная.

Балочный элемент BEAM4, например, имеет 6 степеней свободы (UX, UY, UZ, ROTX, ROTY, ROTZ) в узле и используется для моделирования стержневых конструкций в трехмерном пространстве. Плоский элемент PLANE77 имеет в качестве степеней свободы узловые температуры и может использоваться для моделирования только двумерных объектов.

Для выбора нужного типа элемента необходимо выполнить путь в меню ГИП:

Main Menu > Preprocessor > Element Type > Add/Edit/Delete > Add

или использовать команду в строке ввода:

ET,ITYPE,ENAME.

2.1.5. Определение опций элемента

У каждого типа элементов обычно необходимо задать опции. Эти опции позволяют управлять различными параметрами элемента.

Например, у элемента SOLID95 опции следующие:

– выбор локальной системы координат, связанной с элементом;

– выбор точек, в которых происходит вычисление данных (например напряжений) внутри элемента;

– выбор точек, в которых происходит вычисление данных (например напряжений) на поверхности элемента;

– правило численного интегрирования для построения.

Более подробно о том, какие опции допускает соответствующий тип элемента, необходимо смотреть в разделе помощи по каждому элементу. Для выбора нужных опций элемента необходимо выполнить путь в меню ГИП:

Main Menu > Preprocessor > Element Type > Add/Edit/Delete > Options

или использовать команду в строке ввода:

KEYOPT,ITYPE,KNUM,VALUE.

2.1.6. Определение констант элемента

Для некоторых элементов необходимо задавать константы элемента. В основном константы задаются для элементов, которые используются для моделирования трехмерных моделей сплошной среды моделями низшей размерности, например в случае ферменных, балочных и оболочечных элементов. Константы элемента зависят от типа элемента. Так, например, константы для элемента BEAM3, 2D балочного элемента – это площадь сечения (AREA), момент инерции (IZZ), высота сечения (HEIGHT), константа сдвига (SHEARZ), начальная деформация (ISTRN) и добавленная масса (ADDMAS). Для оболочечных элементов это толщина ТК(I) и др.

Не все элементы требуют определения констант. Более подробно о том, какие константы соответствуют типу элемента, необходимо смотреть в разделе помощи по каждому элементу.

Для задания констант элемента необходимо выполнить путь в меню ГИП:

Main Menu > Preprocessor > Real Constants > Add/Edit/Delete

или использовать команду в строке ввода:

R,NSET,R1,R2,R3,R4,R5,R6.

2.1.7. Определение свойств материала

В зависимости от задачи в ANSYS могут быть заданы следующие свойства материала:

- линейные или нелинейные;
- изотропные, ортотропные и анизотропные;
- зависящие от температуры или независящие.

Для задания свойств материала необходимо выполнить путь в меню ГИП:

Main Menu > Preprocessor > Material Props

или использовать команду в строке ввода:

MP,Lab,MAT,C0,C1,C2,C3,C4.

2.1.8. Создание конечно-элементной модели

Есть два метода создания конечно-элементной модели – это твердотельное моделирование и прямое моделирование. Твердотельное моделирование предполагает, что на первом этапе выполняется создание геометрической модели объекта, т. е. описание его геометрической формы, а затем построение сетки конечных элементов на ней. Прямое моделирование – это непосредственное задание узлов элемента по координатам. Этапы геометрического моделирования и построения сетки рассмотрим ниже на примерах.

2.1.9. Приложение нагрузок

Под нагрузками в ANSYS подразумевается задание всех видов краевых (граничных) условий. Например, в случае решения задачи по механике деформируемого твердого тела это задание поля перемещений на некоторой поверхности (условия закрепления) и поля сил (локальных, поверхностных, объемных). Все нагрузки можно разделить на следующие категории:

- DOF Constraints – ограничения на степени свободы;
- Forces – узловые силы;
- Surface Loads – поверхностные силы;
- Body Loads – объемные силы;
- Inertia Loads – инерционные нагрузки;
- Coupled-field Loads – нагрузки в анализе смешанных полей (термоупругий анализ, аэроупругий анализ и др.).

На этом заканчивается подготовка модели и, соответственно, работа в препроцессоре PREP7. Теперь можно переходить к этапу решения.

2.2. Решение задачи

На этом этапе необходимо использовать процессор решения SOLUTION для того, чтобы определить тип анализа и опции анализа, приложить нагрузки, задать начальные условия и решить задачу.

2.2.1. Определение типа анализа

В ANSYS реализованы следующие типы анализа:

STATIC – стационарный анализ. Используется для решения всех типов задач (механики деформируемого твердого тела, механики жидкости и газа, термического анализа и т. д.).

BUCKLE – анализ задачи устойчивости в линейной постановке. Подразумевает, что предварительно был проведен стационарный анализ с вычислением предварительно напряженного состояния [PSTRES,ON]. Используется только для задач механики твердого деформируемого тела.

MODAL – модальный анализ – анализ конструкции на собственные частоты и формы. Используется только для задач механики твердого деформируемого тела.

HARMIC – гармонический анализ. Используется для задач механики твердого деформируемого тела, механики жидкости и газа и электромагнитного анализа.

TRANS – нестационарный анализ. Используется для решения всех типов задач.

SUBSTR – анализ с применением метода подконструкций. Используется для решения всех типов задач.

SPECTR – спектральный анализ. Подразумевает, что предварительно проведен модальный анализ. Используется только для задач механики твердого деформируемого тела.

Для выбора типа анализа необходимо выполнить путь в меню ГИП:

Main Menu > Preprocessor > Loads > -Analysis Type- New Analysis

или использовать команду в строке ввода:

ANTYPE, Antype, Status.

2.2.2. Спецификация решения

На этом этапе в зависимости от типа выбранного решения, а также в зависимости от типа задачи определяются следующие параметры:

- выбор метода решения получаемых систем уравнений;
- задание параметров решения (шаг нагрузки, количество шагов, шаг интегрирования, количество определяемых собственных форм и др.);
- задание точности решения;
- задание параметров записи результатов в файл и др.

Для корректного задания спецификации решения необходимо знать свойства решений анализируемых задач.

2.2.3. Решение задачи

Чтобы начать решение задачи, необходимо выполнить:

Main Menu > Solution > -Solve- Current LS

или *SOLVE*, или, если дополнительные результаты должны считываться из файла нагрузки:

Main Menu > Solution > -Solve- From LS Files

или *LSSOLVE*.

При выполнении этой команды ANSYS считывает модель и информацию по нагрузкам из файла базы данных и вычисляет результат. При этом результат записывается в файл результатов (*Jobname.RST*, *Jobname.RTH*, *Jobname.RMG*, *Jobname.RFL*), а также и в файл базы данных.

Файлы данных ANSYS

Файл базы данных – это основной файл, образующийся при работе программы. Он имеет расширение *.db*. В файле базы данных программа сохраняет все вводимые данные, результаты решения и постпроцессорной обработки. Главное достоинство этого файла – это возможность быстро модифицировать модель. При работе в любом процессоре результаты сохраняются в одном файле. Сохранить данные в файле можно с помощью меню:

Utility Menu > File > Save as Jobname.DB

или команды *SAVE*. Можно также пользоваться кнопкой *SAVE_DB* в панели инструментов, что рекомендуется перед выполнением сложных операций (булевы операции, построение сетки). Восстановить данные из файла базы данных можно с помощью

Utility Menu > File > Resume Jobname.DB

и далее в открывшемся меню выбрать нужный файл. Очистить содержимое файла базы данных можно с помощью

Utility Menu > File > Clear & Start New.

Изменить имя файла можно с помощью

Utility Menu > File > Change Jobname.

При работе программы образуются файлы с различными расширениями и форматами записи (табл. 2.2).

Таблица 2.2

Файлы, используемые комплексом ANSYS

Тип файла	Расширение	Формат записи
Log-файл	Jobname.LOG	ASCII
Файл ошибок	Jobname.ERR	ASCII
Файл output	Jobname.OUT	ASCII
Файл базы данных	Jobname.DB	Двоичный
Файл результатов: – механический анализ; – термический анализ; – анализ магнитных полей FLOTRAN	Jobname.RST Jobname.RTH Jobname.RMG Jobname.RFL	Двоичный
Файл пошаговой нагрузки	Jobname.Sn	ASCII
Графический файл	Jobname.GRPH	ASCII (специальный формат)
Файл матриц элементов	Jobname.EMAT	Двоичный

Ошибки в работе

При работе с программой могут возникать различные ошибки. ANSYS сигнализирует о них появлением окна сообщения, в котором будет кратко объяснена суть ошибки. В ANSYS все ошибки делятся на критические ошибки (ERROR) и предупреждения (WARNING). Если возникает критическая ошибка, то работа программы остановится и в окне появится сообщение, начинаемое с ERROR. Если возникает предупреждение, то работа программы не

прерывается, а только появляются окна сообщений, начинаемых с WARNING. При работе может возникнуть несколько предупреждений. Они все записываются в файл ошибок Jobname.ERR, который можно просмотреть с помощью

Utility Menu > List > Files > Error File.

2.3. Обзор результатов решения задачи

Как только решение вычислено, можно получить доступ к результатам, используя, в зависимости от типа получаемых данных, два постпроцессора – POST1 и POST26.

Постпроцессор POST1 (General Postprocessor – общий постпроцессор) используется для обзора результатов в стационарной задаче или в течение отдельного шага решения в нестационарной задаче. Этот постпроцессор используется для получения линий уровня напряжений, деформаций, температур и др.:

Main Menu > General Postproc

или */POST1.*

Постпроцессор POST26 (Time History Postprocessor – временной постпроцессор) используется для обзора результатов, зависящих от времени, частоты или другого параметра:

Main Menu > Time Hist Postpro

или */POST26.*

2.4. Общая структура командного файла ANSYS

Итак, исходя из этапов анализа задачи, структура простейшего командного файла будет следующей:

/FILENAME, filename

/TITLE, titlename

/UNITS, units

Блок команд, определяющих переменные, используемые в программе.

Блок команд, определяющих функции, используемые в программе.

/PREP7

Блок команд, отвечающих за построение модели (определение типа элемента, определение опций элемента, определение констант элемента, определение свойств материала, создание конечно-элементной модели, приложение нагрузок).

FINISH

/SOLU

Блок команд, отвечающих за решение (определение типа анализа, спецификация решения, решение задачи).

FINISH

Далее идет постпроцессорная обработка. При этом в зависимости от представления результатов пользуемся одним из двух постпроцессоров.

/POST1

Блок команд общего постпроцессора.

FINISH

/POST26

Блок команд временного постпроцессора.

FINISH

3. ГЕОМЕТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ В ANSYS

Существуют два подхода в геометрическом моделировании в ANSYS: моделирование «снизу вверх» и моделирование «сверху вниз». Основы моделирования построены на геометрической иерархии объектов: объект низшей размерности – точка, и далее по возрастанию – линии, поверхности, объемные тела.

Моделирование «снизу вверх» основано на следующем положении: для создания объекта более высокой размерности необходимо создать объект более низкой размерности.

Моделирование «сверху вниз» основано на конструировании объекта из готовых примитивов (параллелепипед, шар и так далее) с применением булевых операций (сложение, вычитание и другие).

3.1. Моделирование «снизу вверх»

В качестве примера рассмотрим вариант построения высадочной матрицы.

1. Необходимо построить два прямоугольника со следующими данными (табл. 3.1):

Таблица 3.1

Координаты вершины, ширина и высота прямоугольников

	Прямоугольник 1	Прямоугольник 2
WP X (XCORNER)	5	12
WP Y (YCORNER)	0	45
WIDTH	20	13
HEIGHT	45	10

Main Menu > Preprocessor > -Modeling- Create > Areas > Rectangle > By 2 Corners.

Вид меню представлен на рис. 3.1.

При вводе параметров каждого прямоугольника нажимаем Apply, после создания последнего нажимаем ОК (рис. 3.2).

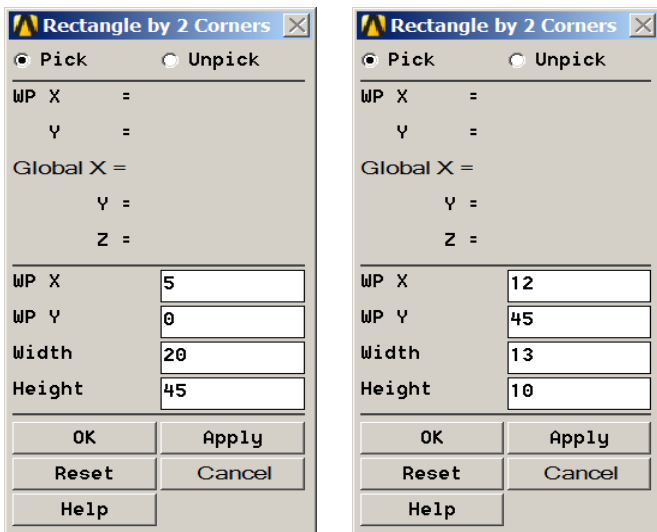


Рис. 3.1. Вид меню

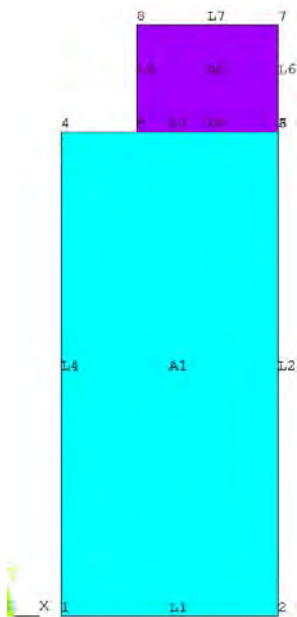


Рис. 3.2. Результат действия

Построим треугольник по точкам 4, 5, 8.

Main Menu > Preprocessor > -Modeling- Create > Areas > Arbitrary > Through KPs.

Вид меню представлен на рис. 3.3.

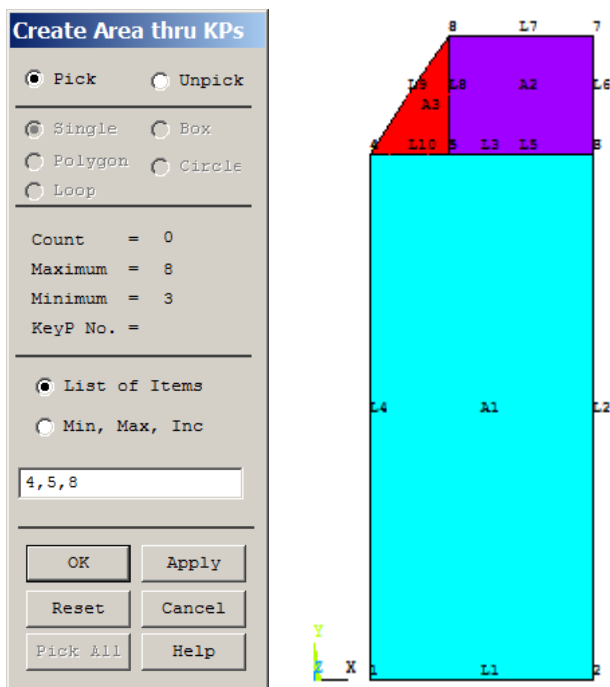


Рис. 3.3. Вид меню и результат действия

Необходимо объединить полученные площади:

Main Menu > Preprocessor > -Modeling- Operate > Booleans > Add > Areas.

Вид меню представлен на рис. 3.4.

Теперь необходимо построить линию, вращением вокруг которой можно получить искомое тело.

Создадим точки с номерами 100 и 101.

Координаты точек линии: (0,0,0) и (0,20,0).

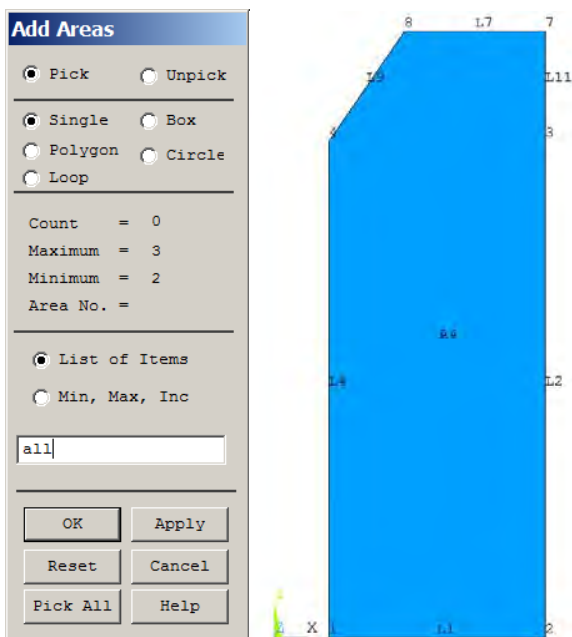


Рис. 3.4. Вид меню и сечение детали

Вид меню представлен на рис. 3.5.

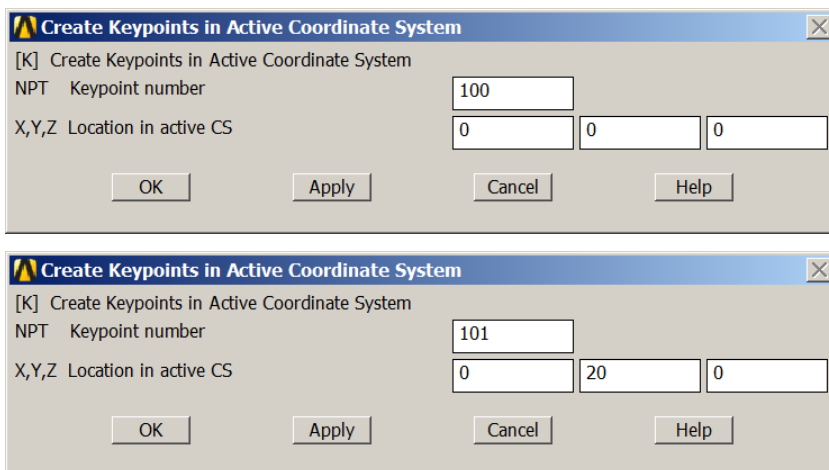


Рис. 3.5. Вид меню

Соединим эти точки отрезком. Вид меню представлен на рис. 3.6.

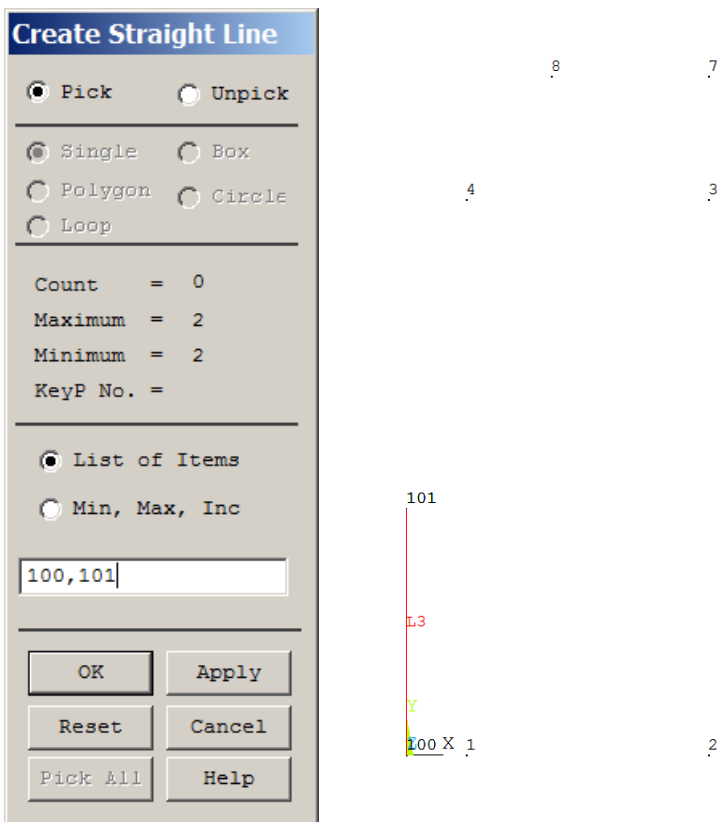


Рис. 3.6. Вид меню и результат действия

Чтобы вновь увидеть площадь, выполним следующее:

Plot > Areas.

Для построения тела вращения необходимо выбрать площадь и ось вращения:

Main Menu > Preprocessor > -Modeling- Operate > Extrude > Areas > About Axis.

Вид меню для выбора тела представлен на рис. 3.7.

LINES

LINE NUM

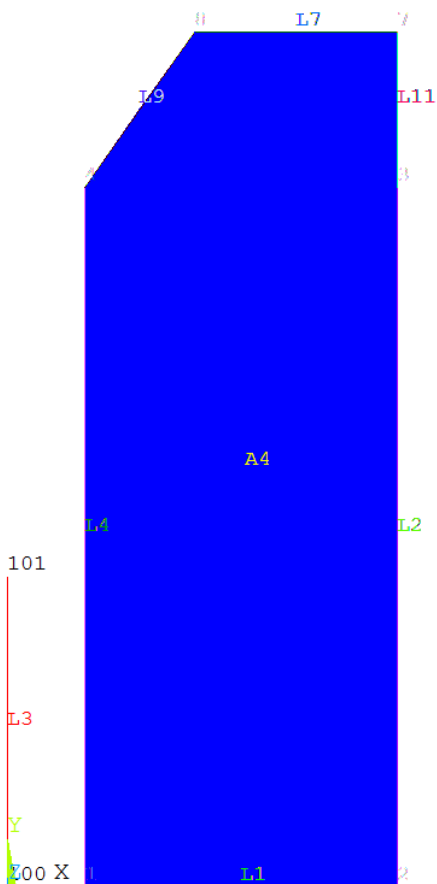
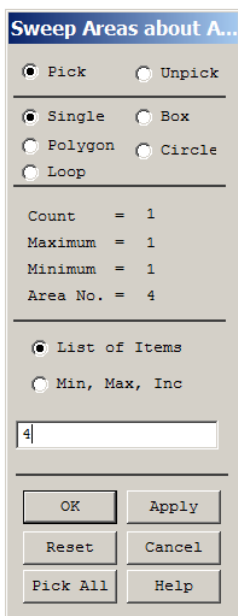


Рис. 3.7. Вид меню и результат действия

Вид меню для выбора оси вращения представлен на рис. 3.8.

В появившемся меню Sweep Areas about Axis (рис. 3.9) нажимаем ОК, однако при необходимости программа позволяет выполнять образование объекта поворотом фигуры на требуемый угол. Вид меню представлен на рис. 3.9.

Итогом проведенных операций является построенная модель матрицы (рис. 3.10).



Рис. 3.8. Вид меню



Рис. 3.9. Вид меню. Диалоговое окно Sweep Areas about Axis

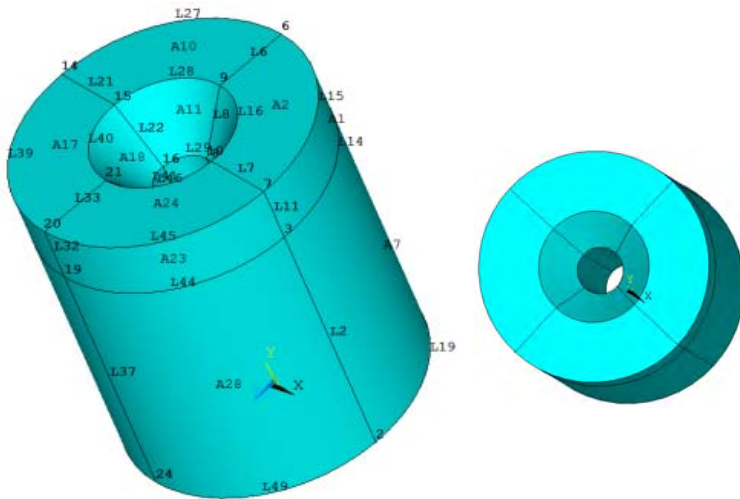


Рис. 3.10. Модель матрицы, построенная методом «снизу вверх»

3.2. Моделирование «сверху вниз»

Создадим два цилиндра со следующими данными (табл. 3.2).

Таблица 3.2

Параметры цилиндров

	Цилиндр 1	Цилиндр 2
WP X (XCORNER)	0	0
WP Y (YCORNER)	0	0
RADIUS	25	5
DEPTH	55	70

Выполним следующее:

*Main Menu > Preprocessor > -Modeling- Create >
Volumes > Cylinder > Solid Cylinder.*

Вид меню представлен на рис. 3.11.

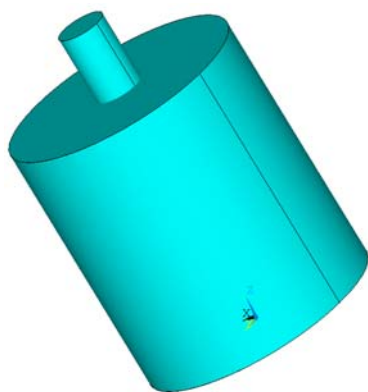
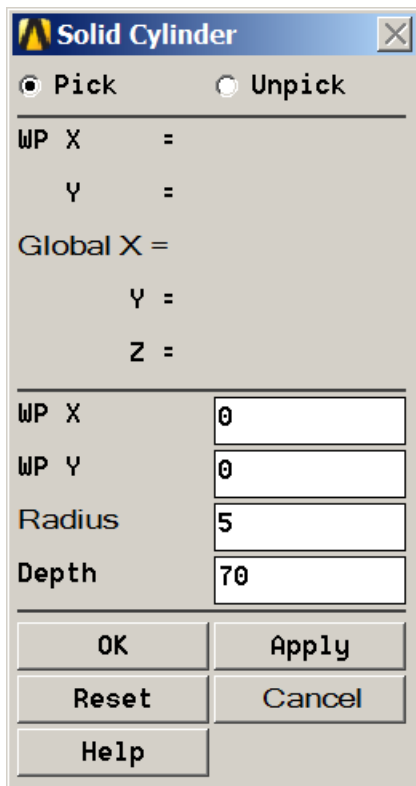
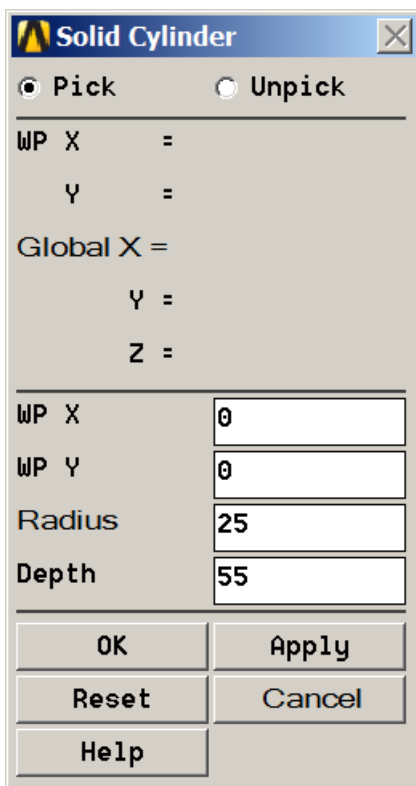


Рис. 3.11. Вид меню и результат действия

Создадим конус со следующими данными (табл. 3.3).

Таблица 3.3

Параметры конуса

WP X (XCORNER)	0
WP Y (YCORNER)	0
RAD-1	12
RAD-2	5
DEPTH	10

Выполним следующее:

*Main Menu > Preprocessor > -Modeling- Create >
Volumes > Cone > By Picking.*

Вид меню представлен на рис. 3.12.

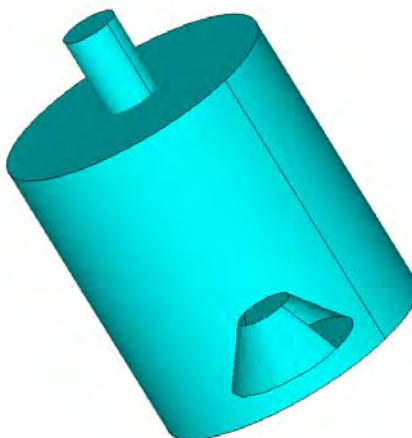
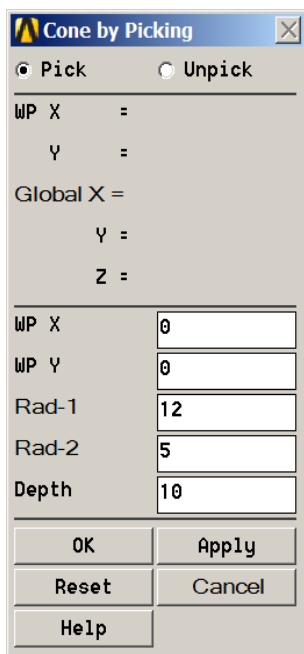


Рис. 3.12. Вид меню и результат действия

Теперь необходимо вычесть из большого цилиндра малый цилиндр и конус. Выполним следующее:

Main Menu > Preprocessor > -Modeling- Operate > Booleans > Subtract > Volumes.

Выделяем цилиндр, из которого производим вычитание, нажимаем Apple.

Вид меню представлен на рис. 3.13.

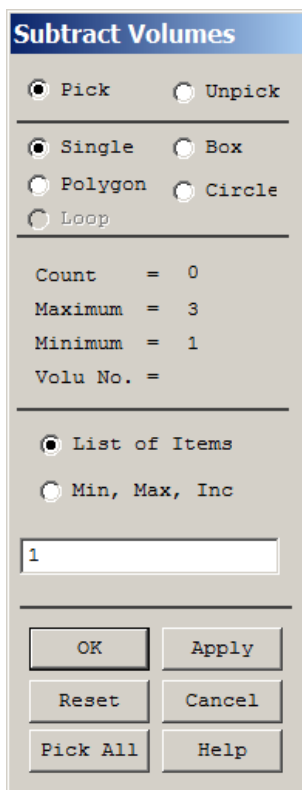


Рис. 3.13. Вид меню

Выделяем цилиндр и конус, которые вычитаем, нажимаем ОК. Вид меню представлен на рис. 3.14.

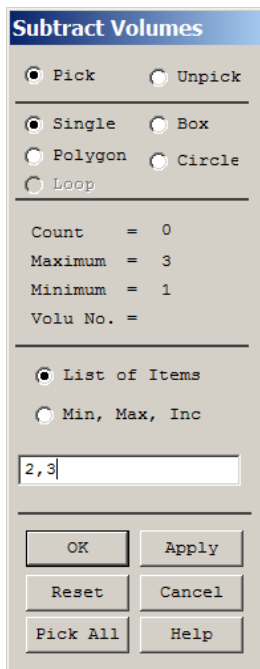


Рис. 3.14. Вид меню

В результате получаем модель матрицы (рис. 3.15).

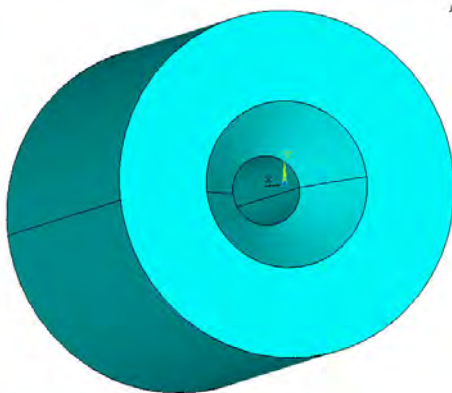


Рис. 3.15. Модель матрицы, построенная «сверху вниз»

4. РАСЧЕТ ПЛОСКОЙ ФЕРМЫ В ANSYS

4.1. Какая конструкция называется фермой

Фермой называется неизменяемая конструкция из прямых стержней, соединенных между собой шарнирами. Силы прикладываются к шарнирам – узлам фермы. Закрепления выполняются также в узлах. В стержнях фермы действуют только продольные (нормальные к поперечным сечениям) внутренние силы.

Статический расчет фермы заключается в определении перемещений узлов, реакций опор, усилий в стержнях, напряжений и деформаций стержней.

Реальная стержневая конструкция с жесткими узлами – сварными, болтовыми или заклепочными соединениями – может рассматриваться как ферма, если размеры узлов значительно меньше расстояний между узлами. В таких конструкциях, как и в фермах, вдали от узлов действуют только продольные внутренние силы. В области жестких узлов картина внутренних сил более сложная и может исследоваться только приближенными методами сопротивления материалов или теории деформированного твердого тела.

Рассмотрим ферму, показанную на рис. 4.1, имеющую n узлов и m стержней, у которой оси стержней и силы лежат в одной плоскости. Перемещения также ограничим этой плоскостью. Такая ферма называется плоской, или двумерной.

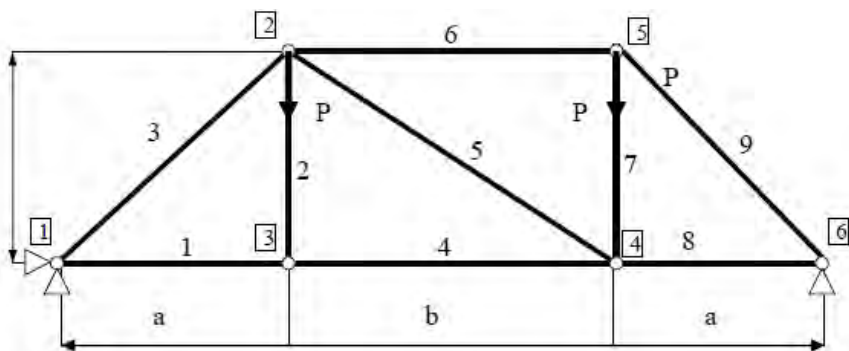


Рис. 4.1. Пример конечно-элементной модели фермы

Ферму сразу можно анализировать как конечно-элементную модель. Стержни являются одномерными конечными элементами, испытывающими только деформацию растяжения-сжатия, их называют ферменными конечными элементами. Шарнирные соединения являются узлами конечно-элементной модели. В качестве степеней свободы берутся узловые перемещения, которые полагаются малыми.

Ферменный (одномерный) конечный элемент показан на рис. 4.2.

С ним связана местная (локальная) система координат (на рисунке символы \bar{x} , \bar{y} подчеркнуты). Общая (глобальная) для всей фермы система координат x , y перенесена параллельным переносом в узел i .

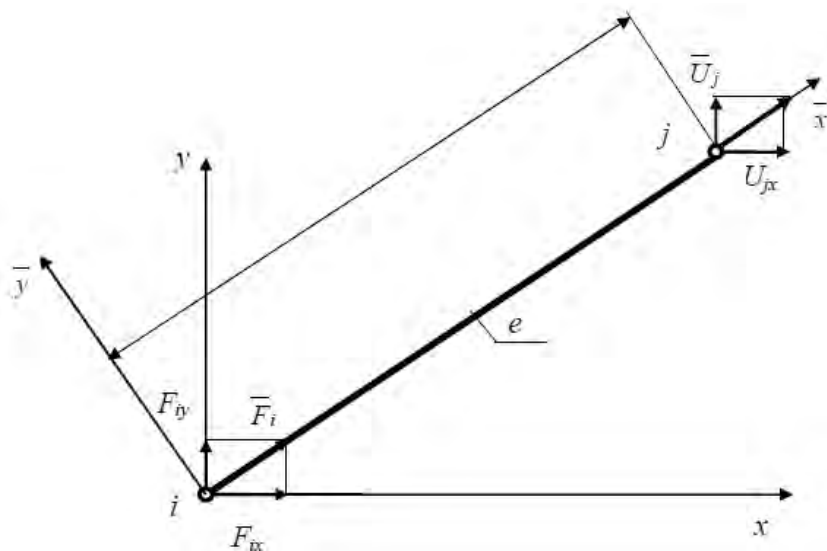


Рис. 4.2. Ферменный (одномерный) конечный элемент

4.2. Пример расчета фермы с использованием графического интерфейса пользователя

В качестве примера расчета фермы рассмотрим задачу № 4.67 (5.7) из задачника [1]. В данной задаче требуется определить опорные реакции и усилия в стержнях фермы, показанной на рис. 4.3, вместе с действующими в узлах силами.

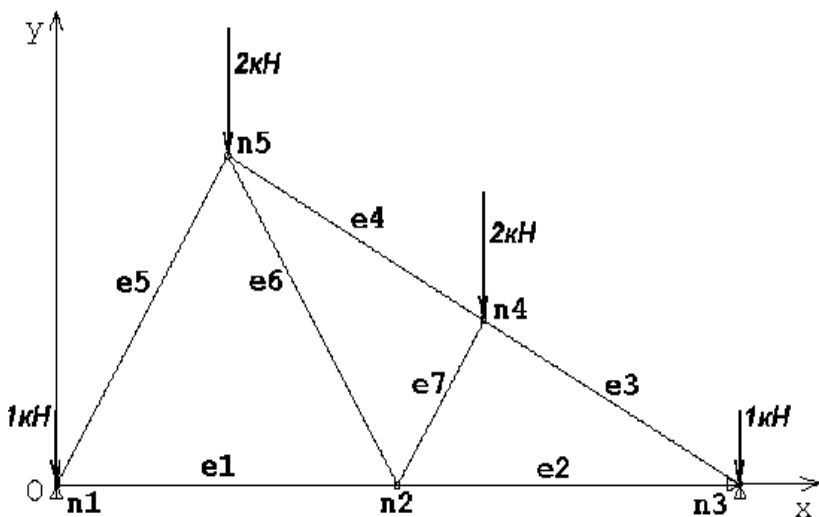


Рис. 4.3. Конечно-элементная расчетная схема фермы

В данном примере рассматривается расчет фермы с использованием графического интерфейса пользователя. Одновременно будут представлены команды ANSYS, позволяющие выполнить те же действия. Ферменная конструкция с точки зрения геометрической модели представляет собой последовательность точек, соединенных линиями.

Для решения представленной задачи в качестве элементов был выбран стержневой конечный элемент LINK1, потому что элементы заданных конструкций не испытывают изгибающих и крутящих моментов, следовательно стержни конструкции испытывают только растяжение или сжатие.

Элемент LINK1 может использоваться в различных инженерных задачах. В зависимости от применения этот элемент может использоваться в качестве фермы, связи, упругого элемента (пружины) и так далее. Двумерный (2D) элемент стержня имеет одну ось, может воспринимать растяжение и сжатие.

Как стержневой элемент, он не имеет свойств изгиба. Трехмерным (3D) элементом стержня является элемент LINK8.

Отнесем ферму к декартовой системе координат Oxy (рис. 4.3) и по рисунку, приведенному в задачнике, рассчитаем координаты ее узлов. На рис. 4.3 точки, где стержни соединены шарнирами,

обозначены номерами $n1, \dots, n5$. Эти точки имеют следующие координаты (в метрах):

$$n1 - (0;0), \quad n2 - (2;0), \quad n3 - (4;0), \quad n4 - (2,5; \sqrt{3}/2), \quad n5 - (1; \sqrt{3}).$$

Для КЭ LINK1 (или LINK8) требуется задать по крайней мере одно материальное свойство (модуль Юнга E) и одну константу КЭ (площадь поперечного сечения AREA).

4.2.1. Построение геометрической части модели

Предварительно в командной строке введем $S3 = \text{SQRT}(3)$, как на рис. 4.4.

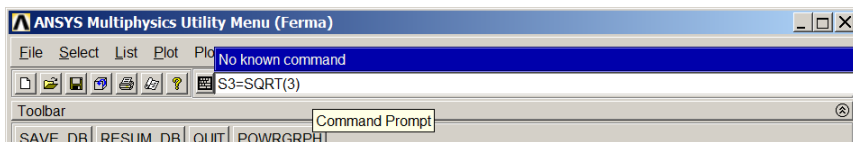


Рис. 4.4. Вид меню

Мы определили значение для скалярного параметра S3. Создадим точки. Выполним следующее:

Main Menu > Preprocessor > -Modeling- Create > Keypoints > In Active CS.

В появившемся окне на первой строке Keypoint number указываем номер точки. Во второй строке Location in active CS вводим координаты точки (табл. 4.1).

Таблица 4.1

Координаты точек

№ точки	X	Y	Z
1	0	0	0
2	2	0	0
3	4	0	0
4	2,5	S3/2	0
5	1	S3	0

При вводе координат точек от 1 до 5 нажимаем кнопку Apply, после введения координат последней точки нажимаем ОК.

Того же можно достигнуть, введя в командной строке

K,1,0,0

K,2,2,0

K,3,4,0

K,4,2.5,S3/2

K,5,1,S3.

или эквивалентную этим командам более короткую запись

K,1,0,0 \$ K,2,2,0 \$ K,3,4,0

K,4,2.5,S3/2 \$ K,5,1,S3.

То есть через знак «\$» можно записывать несколько команд в одной строке.

В графическом окне отобразятся точки с их номерами, как это представлено на рис. 4.5.

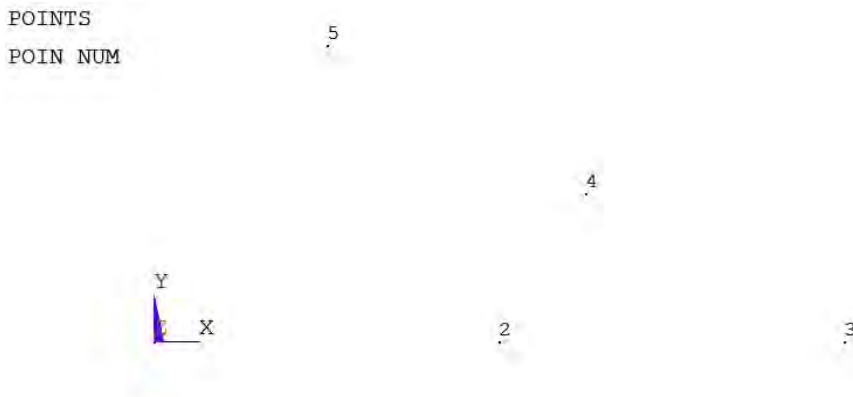


Рис. 4.5. Точки с номерами

Заметим, что для создания отчета рисунок можно сохранить в файле. Выполним следующее:

Utility Menu > Plot Crlts > Hard Copy.

Настройки укажите, как на рис. 4.6.

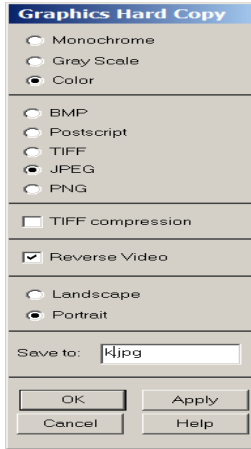


Рис. 4.6. Меню с настройками

Соединим точки линиями. Выполним следующее:

Main Menu > Preprocessor > -Modeling- Create > -Lines- Lines > Straight Line.

Выделяем точку (щелчок мышью), направляем указатель мышки к другой точке и щелкаем на ней. При этом появится линия. Попарно выделяя необходимые точки, построим линии. При появлении линий нажимаем кнопку Apply, после появления последней линии нажимаем OK. Результат отображен на рис. 4.7.

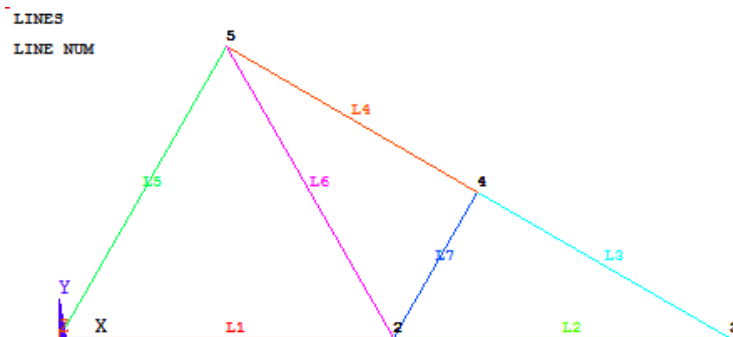


Рис. 4.7. Плоская ферма. Геометрическая часть модели

Того же можно достигнуть, введя в командной строке

L,1,2 \$ L,2,3 \$ L,3,4 \$ L,4,5

L,5,1 \$ L,5,2 \$ L,2,4.

4.2.2. Выбор типа конечного элемента

Выполним следующее:

*Main Menu > Preprocessor > Element Types >
Add/Edit/Delete... > Add.*

В появившемся окне выбираем тип элемента. Нажимаем ОК. В окне Element Types нажимаем кнопку Close.

В этом примере мы поступим несколько иначе: в командной строке введем

ET,1,LINK1.

Это связано с тем, что элемент LINK1 удобно использовать для моделирования плоских ферм. Однако в последних версиях ANSYS описание этого типа элемента отсутствует, хотя программа поддерживает работу с ним. Описание элемента можно найти, например, в файле помощи по ANSYS10.

Установим константы элемента. Выполним следующее:

*Main Menu > Preprocessor > Real Constants >
Add/Edit/Delete... > Add > OK.*

В появившемся окне Real constants нажимаем Add. В окне Element type for real constants – ОК. В меню *Real Constants Set Number* в окне Areal вводим площадь сечения балки величиной $1e-4$. Нажимаем *OK > Close.*

Того же можно достигнуть, введя в командной строке

R,1,1e-4.

4.2.3. Задание свойств материала

Выполним следующее:

*Main Menu > Preprocessor > Material Props >
-Material Models- Isotropic > ОК.*

В появившемся окне вводим: модуль Юнга E и коэффициент Пуассона – Poisson's ratio (minor) ν . Модуль Юнга – $2e11$, коэффициент Пуассона – 0,3. Нажимаем ОК.

В данной задаче коэффициент Пуассона фактически использоваться не будет.

Того же для модуля Юнга можно достигнуть, введя в командной строке

MP,EX,1,2E11.

4.2.4. Построение конечно-элементной сетки

Сначала необходимо задать количество элементов вдоль каждой линии. Для расчета ферм достаточно задать один элемент вдоль линии. Выполним следующее:

Main Menu > Preprocessor > Meshing > Mesh Tool.

В появившемся окне Mesh Tool нажимаем кнопку Set в ряду Lines. В Picking Menu нажимаем Pick All. В появившемся окне Element Sizes on Picked Lines в строке No. of element division указываем число разбиений – 1. Нажимаем ОК.

Результат отображен на рис. 4.8.

Того же можно достигнуть, введя в командной строке

*LSEL,all
CM,ALL_LINE,LINE
LESIZE, ALL_LINE , , 1 , , , , 1.*

Далее нажимаем на Mesh и в Picking Menu нажимаем Pick All. Результат отображен на рис. 4.9.

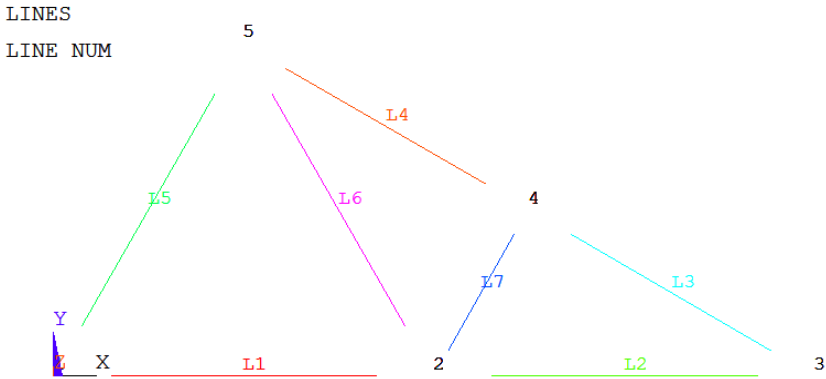


Рис. 4.8. Плоская ферма. На каждой линии один конечный элемент

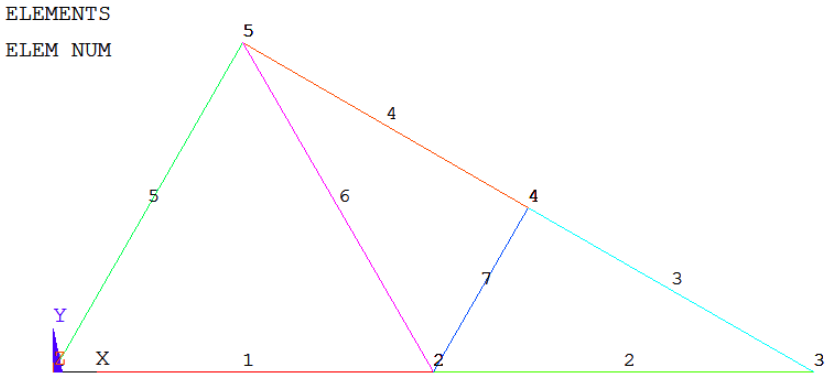


Рис. 4.9. Плоская ферма. Конечно-элементная схема.
Номера элементов и узлов

Того же можно достигнуть, введя в командной строке

LMESH,ALL.

4.2.5. Задание условий закрепления конструкции

Для того чтобы корректно задать условия закрепления, необходимо знать, сколько степеней свободы необходимо закрепить и сколько степеней свободы имеется в узле элемента выбранного типа.

У данного LINK1 две степени: это продольные перемещения первого узла (узел I) и второго узла (узел J). Таким образом, в глобальной системе координат в каждом узле имеются два перемещения UX, UY, соответствующие проекциям на направления глобальных осей (рис. 4.2). Выполним следующее:

*Main Menu > Solution > Define Loads > Apply >
Structural > Displacement > On Nodes.*

Открывается Picking Menu. Выделяем мышью правый узел фермы. В Picking Menu выбираем ОК. В появившемся окне Apply U, Rotonnodes выбираем нужное направление закрепления (в нашем случае UY, UX) и нажимаем ОК. Повторяем операции с нижним левым узлом фермы. Здесь направление закрепления UY.

Результат отображен на рис. 4.10.

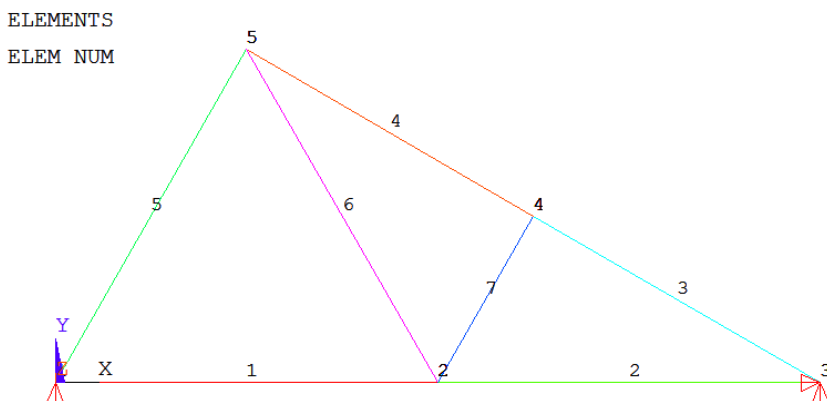


Рис. 4.10. Конечно-элементная схема с заданными закреплениями

Того же можно достигнуть, введя в командной строке

```
D,1,UY $ D,3,UY      ! UY=0 в узлах 1 и 3
D,3,UX              ! UX=0 в узле 3
```

Обратим внимание на то, что весь текст после знака «!>» до конца строки воспринимается как комментарий.

4.2.6. Задание внешних сил, приложенных к конструкции

Выполним следующее:

Main Menu > Solution > Define Loads > Apply > Structural > Force/Moment > On Nodes.

Открывается Picking Menu. Выделяем мышью пятый узел фермы. В Picking Menu – ОК. В появившемся окне Apply F/M on nodes в выпадающем меню Direction of force/mom задаем нужное силовое воздействие FY.

Во второй строке Value задаем величину силы: $-2e3$. Нажимаем ОК. Аналогично задаем нагрузки в узлах по другим степеням свободы.

Результат отображен на рис. 4.11.

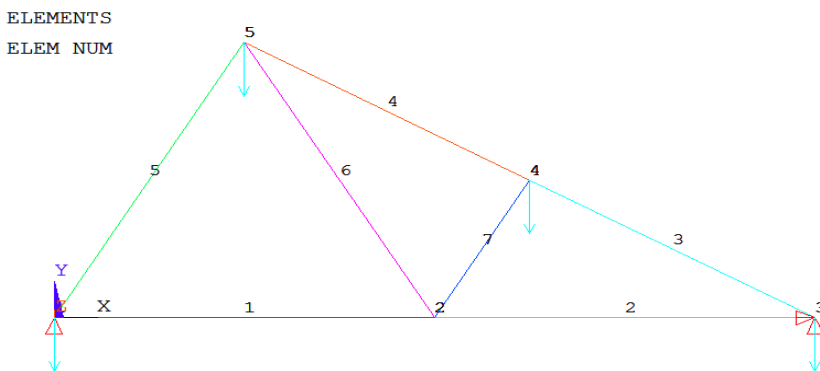


Рис. 4.11. Плоская ферма с приложенными нагрузками и условиями закрепления

Того же можно достигнуть, введя в командной строке

$F,1,FY,-1E3 \ $ F,3,FY,-1E3 \ !FY=-1e3$ в узлах 1 и 3;
 $F,4,FY,-2E3 \ $ F,5,FY,-2E3 \ !FY=-2e3$ в узлах 4 и 5.

Можно при необходимости вывести на экран значения введенных нагрузок в текстовом виде, выполнив действия, представленные на рис. 4.12.

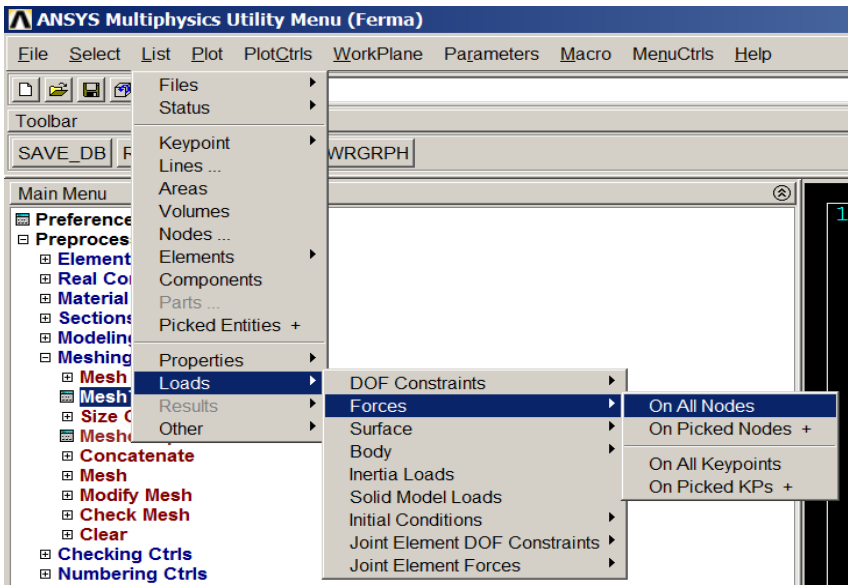


Рис. 4.12. Вид меню

Результат представлен на рис. 4.13.

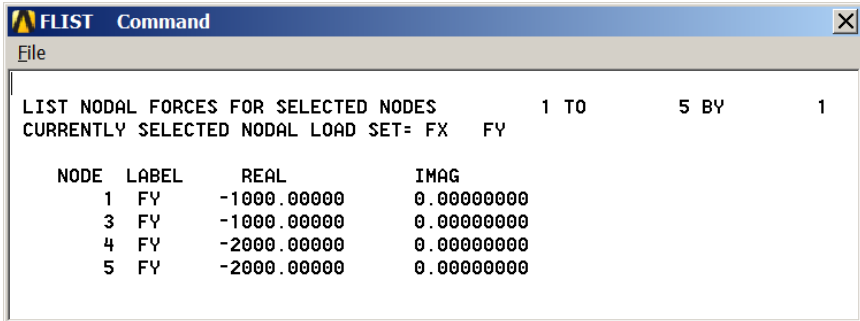


Рис. 4.13. Значения введенных нагрузок в текстовом виде

На этом ввод данных завершается, и остается решить задачу.

4.2.7. Запуск решателя и просмотр результатов

Для запуска решателя необходимо выполнить:

Main Menu > Solution > -Solve- Current LS > OK.

При правильном решении появится окно *Solution is done!* (задача решена).

Того же можно достигнуть, введя в командной строке

```
/SOLU  
ANTYPE,STATIC ! Статический анализ  
SOLVE ! решаем СЛАУ  
FINISH.
```

После решения надо посмотреть деформированное состояние балки.

Для этого выполним:

Main Menu > General Postproc > -Read Results- First Set,

затем

Plot Results > Deformed Shape.

В появившемся окне выбираем вторую строку и нажимаем ОК. Результат представлен на рис. 4.14.

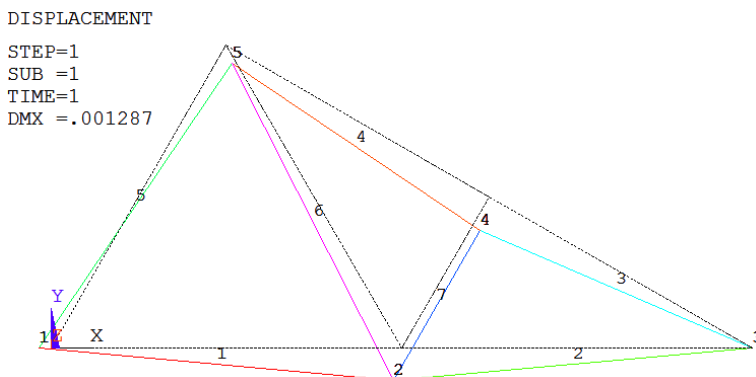


Рис. 4.14. Деформированное и недеформированное состояния плоской фермы

Просмотр эпюр усилий в стержнях.

Для построения эпюр усилий в стержнях необходимо по результатам расчетов создать электронные таблицы данных.

Выполним следующее:

Main Menu > General Postproc > Element Table > Define Table.

В появившемся меню Element Table Data нажать Add. Далее в появившемся меню задаем имя электронной таблицы FI и тип таблицы (SMISC,1) (рис. 4.15).

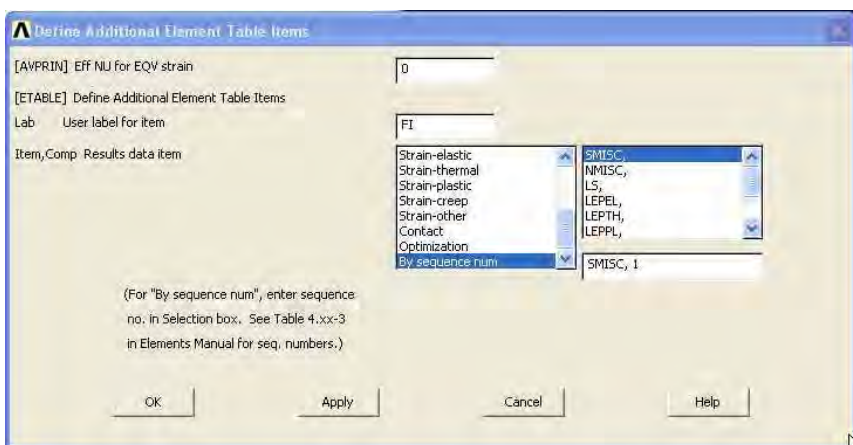


Рис. 4.15. Диалоговое окно Define Additional Table Items

Таким образом, мы создали таблицу с именем FI значений осевых усилий в узлах I (первый узел) конечных элементов.

Повторяем аналогичные действия для создания таблицы с именем FJ значений осевых усилий в узлах J (второй узел) конечных элементов. Закрываем меню.

Выполним следующее:

*Main Menu > General Postproc > Plot Results >
Control Plot > Line Elem Res.*

В появившемся меню выбираем значения, как показано на рис. 4.16.

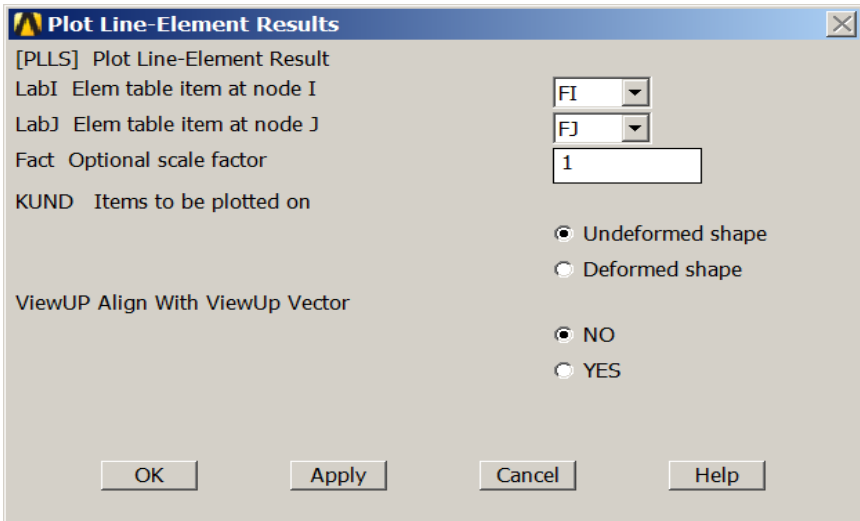


Рис. 4.16. Диалоговое окно Plot Line-Element Results

Нажимаем ОК и получаем эпюры сил (рис. 4.17).

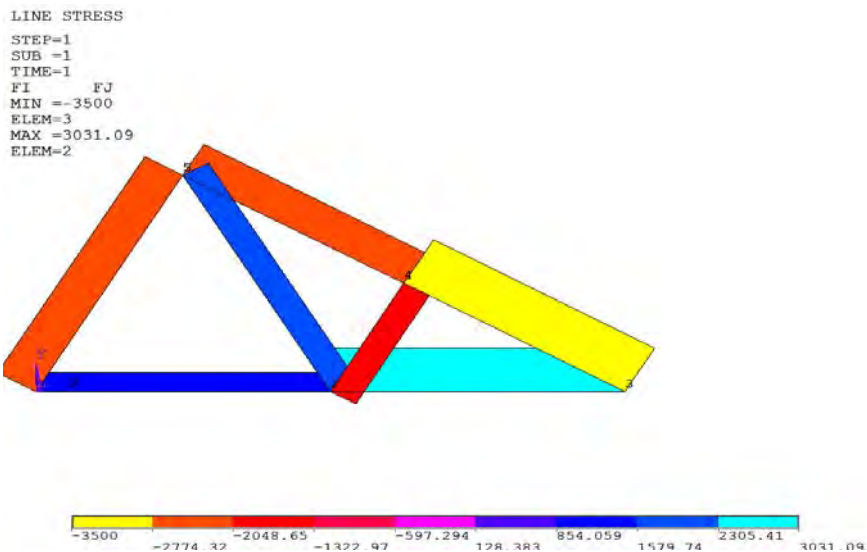


Рис. 4.17. Эпюры осевых сил

Для вывода реакций в текстовом виде выполним следующее:

Main Menu > General Postproc > List Results > Reaction Solu.

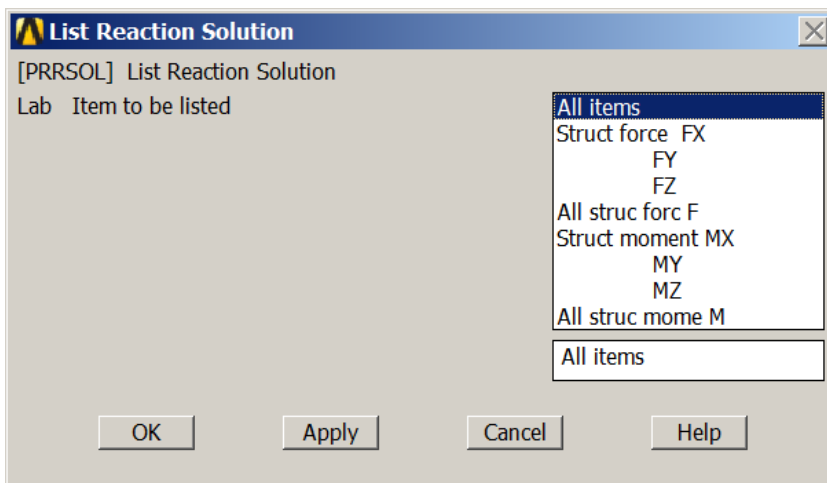


Рис. 4.18. Вид меню для текстового вывода значений реакции в узлах

Результат отображен на рис. 4.19.

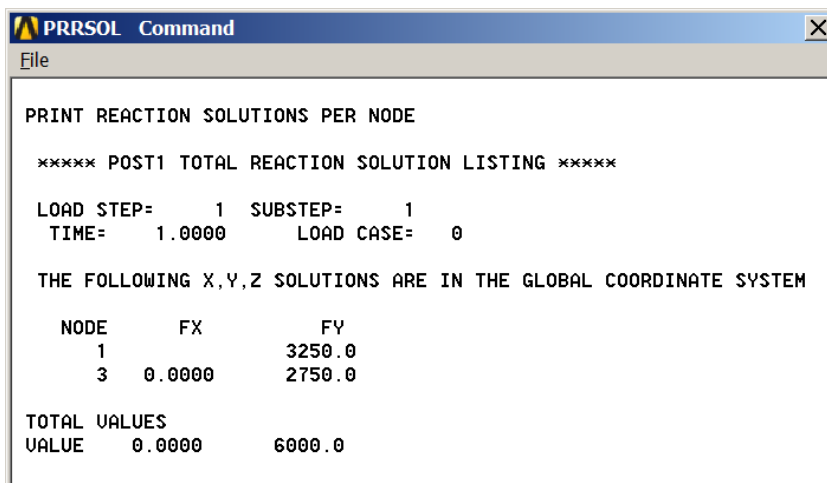


Рис. 4.19. Значения реакций

Того же можно достигнуть, введя в командной строке

PRRSOL ! Печатаем опорные реакции.

Для вывода в текстовом виде значений осевых сил выполняем следующее:

Main Menu > General Postproc > List Results > Element Table Data

и выбираем *FI*, как показано на рис. 4.20.

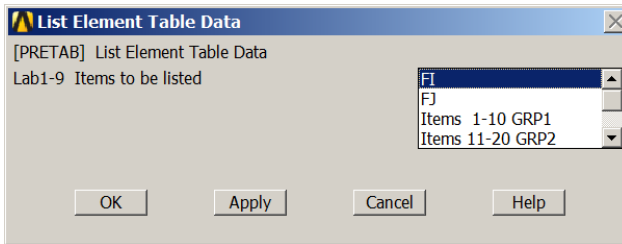


Рис. 4.20. Вид меню

Результат отображен на рис. 4.21.

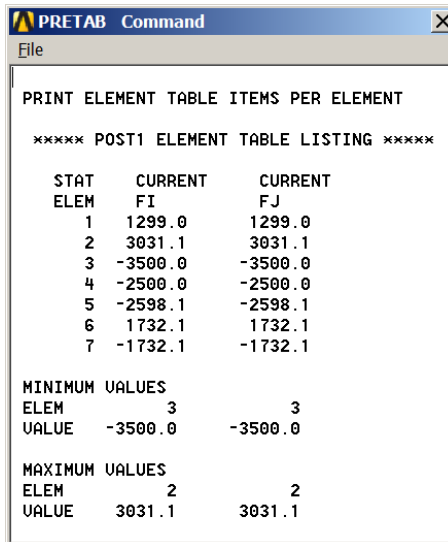


Рис. 4.21. Значения осевых сил

Того же можно достигнуть, введя в командной строке

PRESOL,SMISC,1! Печатаем усилия в стержнях.

Для вывода в текстовом виде всех результатов на всех элементах выполняем следующее:

Main Menu > General Postproc > List Results > Element Solution

и выбираем *Line Element Results > Element Results*, как показано на рис. 4.22.

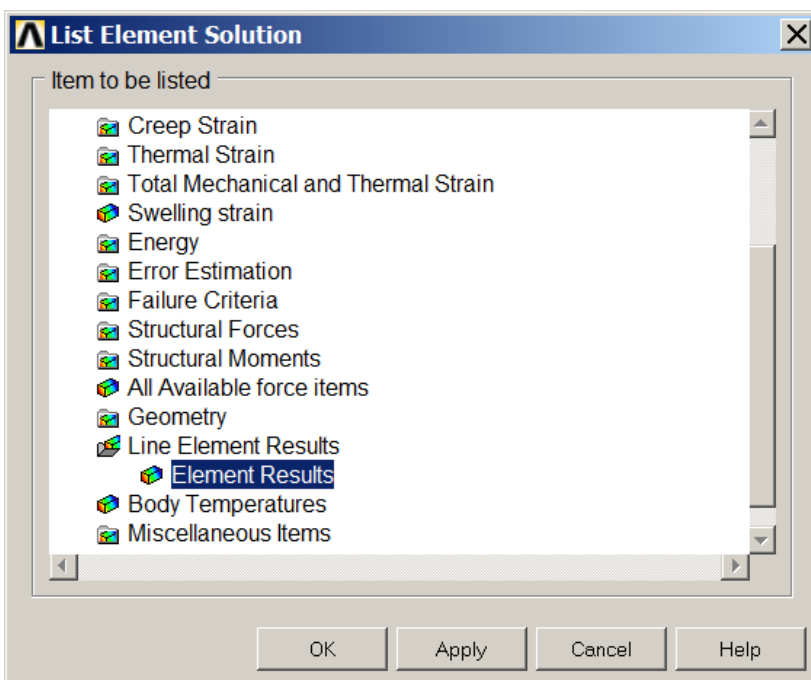


Рис. 4.22. Вид меню

Результат отображен на рис. 4.23.

Того же можно достигнуть, введя в командной строке

PRESOL,ELEM.

```

PRESOL Command
File
PRINT ELEM ELEMENT SOLUTION PER ELEMENT
***** POST1 ELEMENT SOLUTION LISTING *****
LOAD STEP      1 SUBSTEP=      1
TIME=          1.0000      LOAD CASE=  0
EL=            1 NODES=       1      2 MAT=   1
LINK1
TEMP =         0.00      0.00 FLUENCES = 0.000E+00 0.000E+00
MFORX= 1299.0
SAXL= 0.12990E+08 EPELAXL= 0.000065 EPTHAXL= 0.000000 EPSMAXL= 0.000000 EPINAXL= 0.000000
EL=            2 NODES=       2      3 MAT=   1
LINK1
TEMP =         0.00      0.00 FLUENCES = 0.000E+00 0.000E+00
MFORX= 3031.1
SAXL= 0.30311E+08 EPELAXL= 0.000152 EPTHAXL= 0.000000 EPSMAXL= 0.000000 EPINAXL= 0.000000
EL=            3 NODES=       3      4 MAT=   1
LINK1
TEMP =         0.00      0.00 FLUENCES = 0.000E+00 0.000E+00
MFORX= -3500.0
SAXL= -0.35000E+08 EPELAXL= -0.000175 EPTHAXL= 0.000000 EPSMAXL= 0.000000 EPINAXL= 0.000000
EL=            4 NODES=       4      5 MAT=   1
LINK1
TEMP =         0.00      0.00 FLUENCES = 0.000E+00 0.000E+00
MFORX= -2500.0
SAXL= -0.25000E+08 EPELAXL= -0.000125 EPTHAXL= 0.000000 EPSMAXL= 0.000000 EPINAXL= 0.000000
EL=            5 NODES=       5      1 MAT=   1
LINK1
TEMP =         0.00      0.00 FLUENCES = 0.000E+00 0.000E+00
MFORX= -2598.1
SAXL= -0.25981E+08 EPELAXL= -0.000130 EPTHAXL= 0.000000 EPSMAXL= 0.000000 EPINAXL= 0.000000
EL=            6 NODES=       5      2 MAT=   1
LINK1
TEMP =         0.00      0.00 FLUENCES = 0.000E+00 0.000E+00
MFORX= 1732.1
SAXL= 0.17321E+08 EPELAXL= 0.000087 EPTHAXL= 0.000000 EPSMAXL= 0.000000 EPINAXL= 0.000000
EL=            7 NODES=       2      4 MAT=   1

```

Рис. 4.23. Все результаты по элементам

4.2.8. Анализ результатов решения

Анализируя полученные результаты, видим, что максимальная сжимающая нагрузка (NFORX) возникает на элементе с номером 3 и равна -3500 Н, а максимальная растягивающая нагрузка возникает на элементе с номером 2 и равна $3031,1$ Н. Поскольку площадь всех элементов одинакова и равна 10^{-4} м^2 , то максимальные сжимающие напряжения (SAXL) возникают на элементе с номером 3 и равны $-3500 \text{ Н} / 10^{-4} \text{ м}^2 = -0,35 \cdot 10^8 \text{ Н/м}^2$, а максимальные растягивающие напряжения (SAXL) возникают на элементе с номером 2 и равны $3031,1 \cdot 10^{-4} \text{ Н/м}^2 = 0,30311 \cdot 10^8 \text{ Н/м}^2$.

5. АВТОМАТИЗАЦИЯ РАБОТЫ В ANSYS

Процесс решения различных инженерных задач практически всегда связан с созданием геометрии исследуемого объекта. Геометрические модели могут быть достаточно сложными, содержать большое количество элементов, но при этом иметь незначительные в рамках одного типа моделей отличия: количество определенных стандартных компонентов, размеры основных элементов и тому подобные характеристики. В таких случаях было бы целесообразно использовать некоторую шаблонную модель, на основе которой строить геометрию всех остальных ей подобных моделей или, иными словами, автоматизировать процесс создания геометрии модели. Такая автоматизация способствует сокращению затрат времени на исследования свойств модели, что очень важно при большом объеме вычислительной и исследовательской работы в современных конструкторских бюро.

Различные конечно-элементные программные пакеты предлагают несколько путей решения проблемы подобного рода.

Так, в инженерном программном комплексе ANSYS пользователю предлагается автоматизировать процесс создания модели описанными ниже способами:

– *Batch-файл*. Это регистрационный командный файл, выполняющийся в пакетном режиме. Этот файл может быть получен на основе командного файла, который хранится в том же рабочем каталоге, что и файлы проекта, под тем же именем, что и имя проекта, и имеет расширение .log. В этот файл записываются все команды пользователя ANSYS со всеми их параметрами. Имена команд и параметры команд записаны так, что с помощью любого текстового редактора их можно предварительно редактировать, и в отредактированной форме использовать этот файл в качестве входного командного файла для ANSYS. С помощью командного файла можно автоматизировать не только построение геометрии, но и задавать граничные условия, внешние силовые факторы. Чтобы использовать batch-файл, нужно вызвать меню *ANSYS Utility Menu > File > Read Input* и указать имя файла, который следует считывать ANSYS для получения входных команд и их параметров.

– *Mac-файл*. Это файл с наборами команд ANSYS и макрокоманд встроенного в комплекс ANSYS макроязыка APDL. *APDL* –

программный параметрический язык, который встроен в ANSYS в виде интерпретатора. При запуске макроса используется механизм передачи параметров, позволяющий передать некоторые аргументы внутрь макропрограммы. Данное средство предоставляет все возможности по использованию команд и инструментов ANSYS и реализует концепцию автоматизированного программного управления процессами ANSYS. Мас-файл можно создать в любом текстовом редакторе, используя команды ANSYS и APDL. Для вызова макрофайла необходимо вызвать меню *ANSYS Utility Menu > Macro > Execute Macro*, указать имя макро-файла, его входные параметры (в строго определенном порядке).

В некоторых случаях невозможно решить задачу по автоматизации построения той или иной модели без средств программирования. Необходимость в использовании инструментов программирования может возникать в тех ситуациях, когда количество многократно встречающихся в данной модели типовых элементов изначально неизвестно или когда наличие или отсутствие того или иного компонента определяется характеристиками модели, которые, в свою очередь, задаются пользователем при запуске приложения.

Начиная работать с программой, пользователи обычно имеют дело с интерактивным режимом – вся работа ведется через меню. Такой вариант работы позволяет получить доступ к большинству функций и для большинства задач является достаточным. Однако некоторые операции и целые расчеты проще проводить, используя встроенный командный язык.

APDL – это аббревиатура для ANSYS Parametric Design Language. APDL – мощный язык, позволяющий создавать параметрические модели и автоматизировать рутинные операции.

Используя APDL, можно:

- вводить размеры, свойства и так далее, используя параметры, а не числа;
- получать информацию из базы модели;
- производить математические операции над параметрами;
- создавать макросы с набором команд;
- создавать параметрические модели.

Большинство операций, которые пользователь выполняет при работе с программой, могут быть заменены командами. Есть два простейших способа знакомства с языком. Во-первых, при нажатии

кнопки *Help* в любом диалоговом окне выводится помощь по данной операции, включая описание соответствующей команды.

Например, рассмотрим создание точки. В окне диалога можно ввести ее номер и три координаты, рис. 5.1.

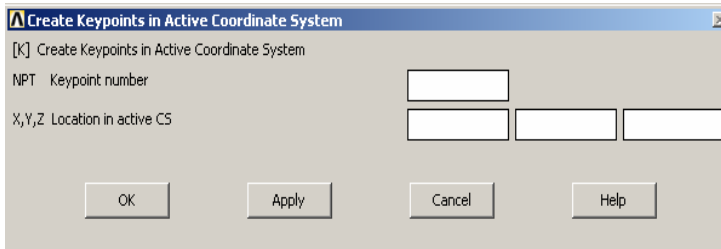


Рис. 5.1. Меню создания точки

При нажатии кнопки *Help* выводится описание команды, рис. 5.2.

K, *NPT*, *X*, *Y*, *Z*

Defines a keypoint.

[PREP7: Keypoints](#)

MP ME ST PR PRN <> <> FL EM <> <> PP <>

NPT

Reference number for keypoint. If zero, the lowest available number is assigned [[NUMSTR](#)].

X, *Y*, *Z*

Keypoint location in the active coordinate system (may be R, θ , Z or R, θ , Φ). If J=P, graphical picking is enabled and all other fields (including *NPT*) are ignored (valid only in the GUI).

Рис. 5.2. Подсказка к команде

Таким образом, команда создания точки выглядит следующим образом:

K,номер,координата x,координата y,координата z.

Номер является необязательным параметром.

Второй способ состоит в следующем. Во время работы пользователя все команды заносятся в файл «имя работы».log. Это текстовый файл, который можно смотреть в любом текстовом редакторе. Если пользователь создал точку номер 5 с координатами 1, 2, 3, то в файле появится команда **K,5,1,2,3**.

По мере работы с программой пользователь может изучать команды, которые он чаще использует.

Команды можно передать в программу двумя способами. Первый способ подходит для ввода единичных команд – они вводятся в окне ввода наверху графического окна, рис. 5.3.

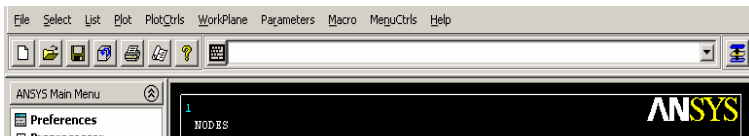


Рис. 5.3. Окно ввода команд

Второй способ позволяет вводить наборы команды из командного файла (или макроса), созданного заранее, *File > Read Input from...* При таком вводе команд можно создавать сложные программы, располагающиеся в нескольких файлах с циклами и ветвлениями. Для полного раскрытия возможностей APDL желательно иметь навыки написания программ на каком-либо языке программирования.

Используя APDL, можно организовать работу с параметрами. Для задания скалярного параметра используется команда *Имя = Значение* или диалоговое окно *Parameters > Scalar parameters...*, рис. 5.4.

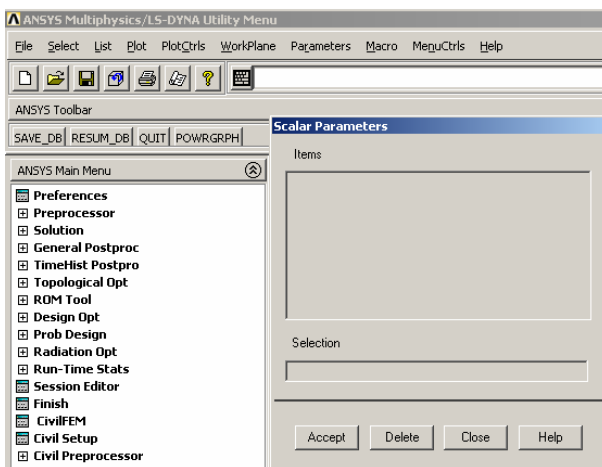


Рис. 5.4. Диалог задания и просмотра скалярных параметров

Для использования параметра необходимо просто ввести его имя вместо числа в диалоге. Например, прямоугольник с параметрами $w = 10$ и $h = 5$ (рис. 5.5) задается следующим образом:

Main Menu > Preprocessor > Modeling > Create > Area > Rectangle > By 2 Corners.

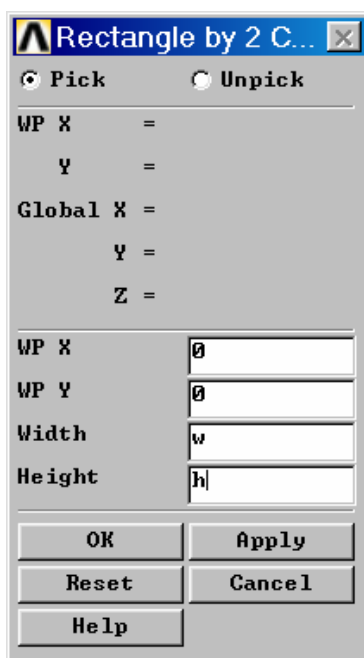


Рис. 5.5. Меню создания прямоугольника

Того же можно достигнуть, введя в командной строке

blc4,,,w,h

(параметры w и h должны быть заданы ранее).

Команды можно использовать как при подготовке модели, так и при настройке и запуске решателя, анализе результатов.

В некоторых случаях возможно написание командных файлов, проводящих полный цикл расчета от создания модели до вывода результатов. За счет использования параметров можно создавать

параметрические модели и использовать их для проведения много-вариантных расчетов или оптимизации.

Например, команда ветвления начинается с **IF* и заканчивается **ENDIF*, можно использовать команды **ELSEIF* и **ELSE* внутри конструкции:

```
*if, x, eq, y, then!если x равно y  
...  
*elseif, x, eq, z, then!иначе если x равно z  
...  
*else !иначе  
...  
*endif
```

Простейший цикл организуется с помощью команд **DO* или **DOWHILE* в начале и **ENDDO* в конце. Например:

```
*do, i, 1, num !цикл от 1 до num по переменной i  
...  
*enddo
```

5.1. Общие сведения о языке APDL

APDL представляет собой параметрический проектный язык для ANSYS. В то время как команды ANSYS могут быть использованы как часть языка описания модели, команды APDL характеризуются большим разнообразием возможностей для программирования, такими, например, как повторение команд, создание и запуск макросов, условий «если – то», циклов, операций над скалярными, векторными и матричными переменными.

APDL можно использовать для решения сложных задач, например для проектной оптимизации и адаптивного слияния.

APDL можно применять:

1. *Для работы с панелью инструментов*: можно добавить часто используемые функции ANSYS или макросы на панель инструментов, определяя для них краткие названия.

2. *Для работы с параметрами*: параметры являются переменными APDL. Используются два типа параметров: скаляр и массив.

3. Как язык макроописаний: можно записать часто используемую последовательность команд ANSYS в макрофайлы (иногда называемые командными файлами). Создание макросов позволяет создавать собственную макрокоманду дополнительно к существующим. Макросы могут вызывать функции графического интерфейса пользователя или оперировать аргументами.

5.2. Переменные в APDL и ANSYS

Как любой язык программирования (макропрограммирования), APDL содержит набор встроенных типов данных и операций, определенных над этими типами.

Переменные в терминологии APDL называются *параметрами*. Тип параметра не нужно явно объявлять. Все числовые величины (независимо от их логического смысла – целые или вещественные) сохраняются как вещественные числа двойной точности (18–20 знаков после запятой в нормализованном виде).

Чтобы объявить параметр, достаточно просто указать его имя в тексте командного файла, в любом математическом выражении или команде ANSYS/APDL. Имя переменной может состоять из любых буквенно-цифровых символов, включать знак подчеркивания, не должно содержать пробелов, знаков пунктуации и командных символов, операторов APDL, начинаться с буквенного символа. Регистр букв не имеет значения.

Замечание: чтобы результат не был неопределенным, в первый раз желательно в явном виде задать его значение. Для этого следует использовать параметр слева от знака присваивания (=, оператор APDL). Параметр, который используется программистом, но не инициализирован оператором присваивания (например, если имя параметра просто указано в пустой строке), автоматически получает значение «мало отличного от нуля» параметра – приблизительно 2^{-100} .

Пример (выдержка из mac-файла):

B
 $A = B.$

В результате параметр B не определен, параметр A определен как $A = B$, следовательно $A = 2^{-100}$. Этот пример также отражает тот факт, что в ANSYS пустые поля в числовых выражениях, где ожидается имя переменной, трактуются как значения, бесконечно близ-

кие к нулю (а при подстановке в качестве фактических параметров в функции – равными нулю).

Хотя интерпретатор ANSYS выдает предупреждения о наличии неинициализированных параметров, в некоторых сложных случаях (при наличии вложенных циклов, сложных условных выражений и т. д.) такое сообщение может быть не выдано. А неинициализированный параметр может служить источником ошибок (порождать деление на ноль, блокировать выполнение циклов и т. д.), которые используют параметр в качестве источника порога итераций. Поэтому следует внимательно следить за инициализацией переменных.

Еще одним видом данных является массив.

Массив – непрерывный набор параметров. Обращение к определенному параметру осуществляется указанием имени массива и индекса параметра (индексация начинается с единицы).

Массивы объявляются специальной командой APDL *DIM.

Массивы могут иметь разную размерность 1D, 2D, 3D, 4D, 5D.

Размерности массивов можно трактовать следующим образом:

1D массивы – строковые массивы. A(5) – обращение к пятому элементу строки-массива;

2D массивы – массивы из строк и столбцов. A(2,5) – обращение к пятому элементу второй строки;

3D массивы – массивы из плоскостей, строк и столбцов. A(4,1,8) – обращение к четвертой плоскости, первой строке, восьмому столбцу;

4D массивы – массивы из строк, столбцов, плоскостей и книг. A(3,2,1,6) – обращение к третьей книге, второй плоскости, первой строке, шестому столбцу;

5D массивы – массивы из строк, столбцов, страниц, книг и полок. A(1,4,3,2,8) – обращение к первой полке, четвертой книге, третьей странице, второй строке, восьмому столбцу.

APDL предоставляет три основных типа массивов:

1. *Символьный массив* – массив, состоящий из наборов символьных элементов. Каждый такой набор не превышает восемь символов в длину.

2. *Строка* – используется, чтобы вводить символьные строки большой длины.

3. *Таблица* – специальный тип числового массива, который используется системой ANSYS, чтобы вычислять неузловые величины (посредством линейной интерполяции).

Для пользовательских массивов установлен лимит используемой памяти, не превышающий величину $2^{31} - 1$ байт. Учитывая это, нетрудно подсчитать, например, что для массива вещественных чисел двойной точности, каждый элемент которого имеет размер 8 байтов, предел числа элементов составляет $(2^{31} - 1) / 8 = 268\,435\,455$ элементов.

Однако следует помнить, что реально эта цифра еще меньше, поскольку графический интерфейс ANSYS, его программная оболочка и решатель так же требуют значительных ресурсов памяти, это значение может быть значительно меньше.

Замечание: для удобства пользования, сокрытия промежуточных данных и высвобождения неиспользуемой памяти («очистки мусора») предусмотрены соответствующие механизмы.

Чтобы удалить ненужный в дальнейшем параметр, можно воспользоваться следующей командой <имя параметра>= (оставить пустое поле после оператора «равно»):

$A = !$ уничтожение параметра A

Как уже указывалось, пустые поля служат указанием об использовании нуля по умолчанию, приведенная выше команда не эквивалентна следующей команде:

$A = 0$

Приравнивание к нулю не уничтожает параметр, а конструкция $A =$ интерпретируется именно как «уничтожить параметр», а не приравнять к нулю.

Уничтожение массива производится командой **DIM* с указанием имени массива, но без параметров.

5.3. Описание некоторых команд APDL

Команда **VWRITE,Par1,Par2,Par3,Par4,Par5,Par6,Par7,Par8,Par9,Par10,Par11,Par12,Par13,Par14,Par15,Par16,Par17,Par18,Par19* записывает данные в файл в форматированной последовательности. Файл для записи может быть создан предварительно командой **CFOPEN*. Если это не было сделано, данные записываются в стандартный выходной файл или выходной файл, установленный пользователем (через команды ANSYS). Можно записать вплоть до 19 параметров (или констант) за один вызов команды. Оставляя пу-

стые поля нежелательно, так как все параметры в полях, следующих за пустым полем, будут проигнорированы. В качестве параметров для записи можно использовать скалярные параметры, массивы, строки, символы.

Для вывода значений счетчика можно использовать ключевое слово *SEQU* – тогда в файл для данного поля будет записано значение счетчика.

Пример:

В соответствии с командами

**VWRITE,ValC1,SEQU*

**VWRITE,ValC2,SEQU*

второе поле первой записи будет содержать значение счетчика, равное 1, а второе поле второй записи будет содержать значение счетчика, равное 2.

Если команда присутствует в командном файле, сразу же после ее вызова в следующей строке командного файла следует использовать спецификатор формата. Можно использовать спецификаторы форматов языка C и языка FORTRAN.

При записи данных в файл в формате FORTRAN спецификатор формата указывать не нужно. Без ограничений можно использовать вещественные типы данных. Формат данных нужно указывать для каждого поля данных индивидуально. Спецификатор данных для каждого конкретного типа данных описывается так же, как и в языке FORTRAN. Желательно избегать использования скаляров и массивов целочисленных данных. Символьные строки могут быть не более 8 символов в длину.

При записи данных в файл в формате языка C большинство ограничений снимаются. Можно использовать целочисленные данные со знаком. Символьные строки не имеют ограничения в восемь символов. Описатель данных каждого типа обязательно начинается с пробела. Определены следующие описатели:

%I – целочисленные знаковые данные;

%G – вещественные числа двойной степени точности;

%C – символьные строки, символьная строка завершается символом *%/*.

Если записываемый параметр является массивом, необходимо указать индекс того элемента массива, с которого начинается счи-

тывание. После этого автоматически запускается счетчик записи, увеличивающий значения индекса записываемого элемента массива на единицу. Если при циклическом вызове команды происходит выход индекса текущего элемента массива за допустимый диапазон, в соответствующее поле автоматически записываются нули.

Команда **VWRITE* может быть использована в любом процессоре.

Команда **CFCLoS* закрывает командный файл, открытый командой **CFOPEN*.

Замечание: эта команда парная команде **CFOPEN*. Пара команд **CFOPEN* – **CFCLoS* не должна содержать вложения других пар команд.

Команда **CFCLoS* может быть использована в любом процессоре.

Команда **DIM,Par,Type,IMAX,JMAX,KMAX,Var1,Var2,Var3,CSYSID* создает массив указанной размерности.

Параметры команды:

Par – имя параметра, который будет массивом.

Type – тип массива, может определять:

Array – массив, подобный стандартному линейному массиву, индексы которого – целые числа, начиная с единицы. Используется для создания 1D, 2D или 3D массивов.

ARR4 – то же, что *Array*, но используется для создания 4D массивов.

ARR5 – то же, что *Array*, но используется для создания 5D массивов.

Char – данные массива являются символьными строками (вплоть до 8 символов в каждой строке). Индексные номера – целые числа, начинающиеся с единицы.

Table – таблица, индексы массива – вещественные числа, которые должны быть определены при создании таблицы. Элементам строк и столбцов присвоены значения, близкие к нулю, по умолчанию. Целочисленные индексные номера назначаются автоматически, первый элемент строки, столбца, массива и так далее автоматически имеет индекс 1. Индексы используются для доступа к конкретному элементу массива. При считывании элемента по его вещественному индексу автоматически выполняется интерполяция по соседним узлам массива и возвращается полученное в ее ходе значение. Позволяет создавать 1D, 2D или 3D таблицы.

TAB4 – то же, что *Table*, но используется для создания 4D таблиц.

TAB5 – то же, что *Table*, но используется для создания 5D таблиц.

STRING – данные массива являются символьными строками (длиной до *IMAX* символов каждый). Индексные номера для строк и столбцов начинаются с единицы. Индекс столбца является позицией символа в строке.

IMAX – длина первого измерения массива (количество строк). По умолчанию устанавливается равным 1.

JMAX – длина второго измерения массива (количество столбцов). По умолчанию устанавливается равным 1.

KMAX – длина третьего измерения массива (количество плоскостей). По умолчанию устанавливается равным 1.

Var1 – имя, сопоставляемое первому измерению массива, если параметр *Type = Table*. По умолчанию – *Row*.

Var2 – имя, сопоставляемое второму измерению массива, если параметр *Type = Table*. По умолчанию – *Column*.

Var3 – имя, сопоставляемое третьему измерению массива, если параметр *Type = Table*. По умолчанию – *Plane*.

CSYSID – целое число, соответствующее номеру системы координат ID.

Команда **DIM* может быть использована в любом процессоре.

Команда **IF, val1, oper1, val2, base1, val3, oper2, val4, base2* – команда условного выполнения ветви алгоритма.

Параметры:

val1 – первый числовой параметр (или выражение, результатом вычисления которого будет числовой параметр) в условном действии сравнения. В качестве параметров могут быть и символьные строки.

oper1 – метка операции сравнения. Допуск для операций сравнения числовых величин 10^{-10} . Могут быть использованы следующие метки для проверки:

EQ – равенство ($val1 = val2$);

NE – неравенство ($val1 \neq val2$);

LT – меньше, чем ($val1 < val2$);

GT – больше, чем ($val1 > val2$);

LE – меньше или равно ($val1 \leq val2$);

GE – больше или равно ($val1 \geq val2$);

ABLT – вычисляет абсолютные значения для параметров *val1* и *val2* перед выполнением сравнения $val1 \leq val2$;

ABGT – вычисляет абсолютные значения для параметров *val1* и *val2* перед выполнением сравнения $val1 \geq val2$.

val2 – второй числовой параметр для операции сравнения (или выражение, результатом вычисления которого будет числовой параметр).

base1 – действие, выполняемое в случае истинности первого логического сравнения, заданного параметром *oper1*, выполняемого для *val1* и *val2*. Действие может быть задано метками двух групп. Метки первой группы определяют действия, игнорирующие параметры команды после поля *base1*, метки второй группы указывают метод логической компоновки логических операций, заданных в *oper1* и *oper2*.

Метки первого тина:

:label – переход на определенную пользователем метку. Метка должна начинаться с «:» и быть не более восьми символов в длину. Не следует использовать переход по меткам в конструкциях *do-endo* (циклах) и *if-then-else* – переход внутрь блоков и выход из них с помощью пользовательских меток запрещен.

stop – вызовет выход из программы ANSYS при выполнении ее в пакетном режиме. В диалоговом режиме программа не прекратится.

exit – завершает цикл *do-loop*.

cycle – пропускает текущую итерацию цикла *do-endo*.

then – используется для организации конструкции *if-then-else*.

Метки второго тина:

and – связывает *oper1* и *oper2* логическим оператором «и»;

or – связывает *oper1* и *oper2* логическим оператором «или»;

xor – связывает *oper1* и *oper2* логическим оператором «исключающее или».

val3 – третий числовой параметр для операции сравнения (или выражение, результатом вычисления которого будет числовой параметр);

oper2 – метка операции сравнения применительно к *val3* и *val4*, по сути аналогична *oper1*.

val4 – четвертый числовой параметр для операции сравнения (или выражение, результатом вычисления которого будет числовой параметр);

base2 – то же, что *base1*.

Эта команда используется для условного перехода между блоками команд. Команда может быть использована как внутри блока *do-*

enddo, так и в виде самостоятельной команды для организации ветвления *if-then-else* (если-то-иначе). Допускается последовательное вложение блоков *if* до двадцати уровней.

Пример конструкции **IF* (переход на метку по условию, *if* цикл):

```
C1=9!НАЧАЛЬНОЕ ЗНАЧЕНИЕ C1
C2=15!НАЧАЛЬНОЕ ЗНАЧЕНИЕ C2
C3=7!НАЧАЛЬНОЕ ЗНАЧЕНИЕ C3
C4=2!НАЧАЛЬНОЕ ЗНАЧЕНИЕ C4
:RECALC!МЕТКА
C1=C1+1!УВЕЛИЧИТЬ C1 НА ЕДИНИЦУ
C4=C4+1!УВЕЛИЧИТЬ C4 НА ЕДИНИЦУ
!ПОКА C1<C2 ИЛИ C3≥C4
!ВОЗВРАЩАТЬСЯ НА МЕТКУ :RECALC
*IF,C1,LT,C2,OR,C3,GE,C4,:RECALC
```

В ходе выполнения такого набора команд будет шесть раз выполнен переход на метку *:recalc*, описанную ранее команды **IF*. В результате *c1* примет значение 15, *c4* примет значение 8. Таким образом в этом примере организован *if*-цикл. Команда **IF* может быть использована в любом процессоре.

Команда **ELSEIF, val1, oper1, val2, conj, val3, oper2, val4* – команда условного выполнения ветви алгоритма *if-else*. Команда парная команде **IF*. Ветвь выполняется, если результат выполнения логического выражения команды **IF* принимает значение «ложь», а результат логического выражения команды принимает значение «истина».

Параметры:

val1 – первый числовой параметр (или выражение, результатом вычисления которого будет числовой параметр) в условном действии сравнения. В качестве параметров могут быть и символьные строки.

oper1 – метка операции сравнения. То же, что и в рассмотренной выше команде **IF*.

val2 – второй числовой параметр для операции сравнения (или выражения, результатом вычисления которого будет числовой параметр).

conj – условие, связывающее результаты выполнения логического сравнения, заданного *oper1* и *oper2*:

and – связывает *oper1* и *oper2* логическим оператором «и»;
or – связывает *oper1* и *oper2* логическим оператором «или»;
xor – связывает *oper1* и *oper2* логическим оператором «исключающее или».

val3 – третий числовой параметр для операции сравнения (или выражения, результатом вычисления которого будет числовой параметр);

oper2 – метка операции сравнения применительно к *val3* и *val4*, по сути аналогична *oper1*.

val4 – четвертый числовой параметр для операции сравнения (или выражения, результатом вычисления которого будет числовой параметр).

Команда **ELSEIF* может быть использована в любом процессоре.

**ELSE* – команда-разделитель для команды **IF*.

Используется как разделитель для ветвления в блок *if-then-else*, команды внутри ветви **ELSE* выполняются, если результатом вычисления логического выражения команды **IF* было значение «ложь». Команда **ELSE* может быть использована в любом процессоре.

Команда **ENDIF* указывает на завершение тела конструкции *if-then-else*.

Команда обязательно должна завершать блоки *if*, *if-then*, *if-then-else*, *if-then*. Команда **ENDIF* может быть использована в любом процессоре.

Команда **DO,par,ival,fval,inc* организует цикл, указывает начало тела цикла.

Параметры:

par – имя скалярного параметра, который используется как счетчик итераций цикла. Нельзя использовать строковые параметры;

ival – начальное значение счетчика;

fval – конечное значение счетчика;

inc – шаг счетчика (по умолчанию единица).

Для всех параметров допускается использовать отрицательные числа. Можно использовать нецелочисленные параметры. Если *ival* > *fval*, выполнение цикла прекращается (если указанное имеет место при первом входе в цикл, тело цикла не выполнится ни разу). Команда **DO* может быть использована в любом процессоре.

Команда **ENDDO* указывает на завершение тела цикла.

Эта команда парная команде **DO*. Все команды, расположенные между **DO–*ENDDO* образуют тело цикла. Команда **ENDDO* может быть использована в любом процессоре.

Пример использования блока **DO–*ENDDO* с применением команды **IF* для пропуска итераций:

```
C2=10 !НАЧАЛЬНОЕ ЗНАЧЕНИЕ C2
C3=7 !НАЧАЛЬНОЕ ЗНАЧЕНИЕ C3
!ЦИКЛ ОТ 1 ДО 10 С ШАГОМ 1, НАЧАЛО ТЕЛА ЦИКЛА
*DO,COUNT,1,10,
C2=C2+1!УВЕЛИЧИТЬ C2 НА ЕДИНИЦУ
!ЕСЛИ C3 > C2 – ПРОПУСТИТЬ
!ПОСЛЕДУЮЩИЕ !КОМАНДЫ ЦИКЛА
*IF,C3,GT,C2,CYCLE
C3=C3+2!УВЕЛИЧИТЬ C3 НА 2
*ENDDO!ОКОНЧАНИЕ ТЕЛА ЦИКЛА
```

После выполнения алгоритма оказывается, что $C2 = 20$, $C3 = 21$. Если бы пропуска итераций не было, то параметр $C3$ после выполнения всех итераций принял бы значение 27.

Команда **EXIT* прерывает выполнение цикла, организованного командой **DO*. Команда может быть использована в теле команды условного выполнения **IF*:

```
C2=10 !НАЧАЛЬНОЕ ЗНАЧЕНИЕ C2
C3=7 !НАЧАЛЬНОЕ ЗНАЧЕНИЕ C3
!ЦИКЛ ОТ 1 ДО 10 С ШАГОМ 1, НАЧАЛО ТЕЛА ЦИКЛА
*DO,COUNT,1,10,
C2=C2+1 !УВЕЛИЧИТЬ C2 НА ЕДИНИЦУ
*IF,C3,GT,C2,THEN !ЕСЛИ C3 > C2 –
*EXIT !ВЫЙТИ ИЗ ЦИКЛА
C3=C3+2! УВЕЛИЧИТЬ C3 НА 2
*ENDDO !ОКОНЧАНИЕ ТЕЛА ЦИКЛА
!ПОСЛЕ ВЫЗОВА *EXIT ИМЕННО С ЭТОЙ
! СТРОКИ БУДЕТ
!ПРОДОЛЖЕНО ВЫПОЛНЕНИЕ МАКРОПРОГРАММЫ
*F=C3-F4
```

Команда **EXIT* может быть использована в любом процессоре.

Команда **GET,par,entity,entnum,item1,it1num,item2,it2num* сохраняет указанный параметр, описывающий свойство определенного объекта как скалярный параметр или часть параметра массива.

Параметры:

par – имя параметра для сохранения результата запроса на чтение свойства. Массив должен быть определен заранее;

entity – указывает тип свойств объекта, которые будут сохранены. Можно применять следующие ключевые слова для этого параметра (основные):

ACTIVE – текущий активный компонент;

AREA – параметры и статистика поверхностей;

CDSY – параметры и статистика по координатным системам проекта;

CE – параметры и статистика по определенным в проекте нагрузкам;

CP – параметры и статистика для парных элементов;

EDCC – параметры и статистика для контактных объектов модели;

ELEM – геометрические, физические, механические параметры для выбранного элемента;

ETYP – номер набора свойств указанного элемента;

LINE – параметры и статистика линий;

MPLAB – данные элементов с указанной температурой;

NODE – параметры и статистика узлов;

PART – параметры и статистика частей;

SCTN – геометрические параметры секций;

SECP – статистика элементов секций;

SHEL – параметры и статистика оболочек;

TBFT – параметры и статистика для используемых в проекте наборов свойств материалов;

VOLU – параметры и статистика для объемных примитивов;

entnum – указывает номер компонента или номер совокупности компонентов (совокупность формируется при использовании команд выбора ANSYS), о которых нужно получить данные;

item1 – имя конкретного элемента;

it1num – число (или метка) для определенного элемента (если определен *item1*, для некоторых элементов *item1* не требуется *it1num*;

item2, it2num – вторичный набор описателей элементов (требуется только для специфических команд).

Примеры использования команды:

```
! BCD = номер материала элемента номер 97  
*GET,BCD,ELEM,97,ATTR,MAT  
! V37 = объем элемента номер 37  
*GET,V37,ELEM,37,VOLU  
! EL52 = мощность тепл. источника элемента номер 52  
*GET,EL52,ELEM,52,HGEN  
! OPER = тепл. коэф.на элементе 102, поверхности 2  
*GET,OPER,ELEM,102,HCOE,2  
! TMP = средняя темпер. на элементе 16, поверхности 3  
*GET,TMP,ELEM,16,TBULK,3  
! NMAX = номер макс. активного узла  
*GET,NMAX,NODE,,NUM,MAX  
! HN = мощность тепл. источника в узле с номером 12  
*GET,HN,NODE,12,HGEN  
! COORD = номер активной системы координат  
*GET,COORD,ACTIVE,,CSYS
```

5.4. Командный файл для решения задачи о ферме

Приведем пример командного файла с комментариями для решения задачи о ферме, рассмотренной в разделе 4.

```
/PREP7 !Вход в препроцессор  
ET,1,LINK1 ! Стержневой КЭ LINK1  
!Площадь поперечного сечения стержней AREA=1e-4  
R,1,1e-4  
! Модуль Юнга материалов стержней EX=2e11  
MP,EX,1,2E11  
! Вспомогательная переменная  
S3=SQRT(3)  
! Определяем точки по координатам на плоскости  
K,1,0,0 $ K,2,2,0 $ K,3,4,0  
K,4,2.5,S3/2 $ K,5,1,S3  
! Определяем линии по точкам  
L,1,2 $ L,2,3 $ L,3,4 $ L,4,5  
L,5,1 $ L,5,2 $ L,2,4
```

```

! Селектируем все линии
LSEL,all
! Создаем компоненту с именем ALL_LINE
CM,ALL_LINE,LINE
! На каждой линии этой компоненты будет по
! одному конечному элементу
LESIZE, ALL_LINE , , 1 , , , 1
! Наносим конечно-элементную сетку
LMESH,ALL

! Определяем шарнирные опоры в узлах
D,1,UY $ D,3,UY ! UY=0 в узлах 1 и 3
D,3,UX ! UX=0 в узле 3
! Задаем силы в узлах
! FY=-1e3 в узлах 1 и 3
F,1,FY,-1E3 $ F,3,FY,-1E3
! FY=-2e3 в узлах 4 и 5
F,4,FY,-2E3 $ F,5,FY,-2E3
FINISH !Выход из препроцессора
/SOLU !Вход в решатель
ANTYPE,STATIC ! Статический анализ
SOLVE ! решаем СЛАУ
FINISH!Выход из решателя
/POST1 !Вход в постпроцессор
PRRSOL ! Печатаем опорные реакции
PRESOL,SMISC,1 ! Печатаем усилия в стержнях
!Печатаем все результаты на элементах
PRESOL,ELEM
/OUTPUT
FINISH !Вход из постпроцессора

```

Как отмечалось выше, ферму сразу можно анализировать как конечно-элементную модель. Пример такого подхода приведен в файле SMs1.inp из методического пособия [2]. Основное отличие заключается в том, что геометрическая модель не строится, а сразу определяются узлы по их координатам и элементы по узлам, как это представлено ниже.

! Определяем узлы по координатам на плоскости

N,1,0,0

N,2,2,0

N,3,4,0

N,4,2.5,S3/2

N,5,1,S3

! Определяем элементы текущего типа TYPE=1

!(LINK1) с текущими наборами констант REAL=1

! и материальных свойств MAT=1

! по номерам граничных узлов

E,1,2

E,2,3

E,3,4

E,4,5

E,5,1

E,5,2

E,2,4

6. ВЫПОЛНЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

По образцу рассмотренного примера требуется выполнить моделирование деформированного состояния своего варианта фермы, номер которого задается преподавателем. Схемы ферм для различных вариантов даны в разделе 6.2 (рис. 6.1–6.12). Необходимые параметры для разных вариантов задания приведены в табл. 6.1.

Моделирование выполните двумя способами: с использованием диалогового режима и на основе командного файла.

6.1. Оформление отчета

Отчет должен включать следующие пункты:

1. Постановка задачи для своего варианта:

1.1. Силовая схема (конструкция, закрепления, силы).

1.2. Исходные данные (величины сил, свойства материала).

1.3. Описание величин, которые требуется найти.

2. Короткое описание процедуры моделирования в ANSYS.

3. Результаты решения задачи в ANSYS должны включать:

3.1. Геометрическую модель конструкции с указанием номеров точек и линий.

3.2. Конечно-элементную модель конструкции с указанием номеров узлов и элементов.

3.3. Схему деформированной конструкции, наложенную на схему недеформированной конструкции, с указанием наибольшего смещения.

3.4. На силовой схеме нужно показать распределение реакций и указать величины реакции.

3.5. Таблицу реакций в текстовом виде.

3.6. Таблицу сил и напряжений в стержнях в текстовом виде.

6.2. Задания к расчету

Таблица 6.1

Задания к расчету ферм

№ варианта	№ фермы	Размеры фермы, м					Углы, град		Силы, кН		
		a	b	c	d	l	α	β	P_1	P_2	P_3
1	1	1	1	2			30	45	2	3	4
2	1	2	3	1			60	30	3	2	5
3	2	2	2	3	2		45	60	1	2	3
4	2	1	1	4	3		30	45	3	1	2
5	3	2	2	1	3		60	45	2	1	3
6	3	1	1	1	2		30	60	1	3	2
7	4	2	2	1	1		60	30	1	2	3
8	4	3	1	2	2		30	45	3	4	2
9	5	1	1	1	1	2	45	60	4	2	1
10	5	2	1	1	2	1	60	30	3	3	2
11	6	2	2	1	1		30	45	2	1	1
12	6	3	3	2	1		45	60	1	2	2

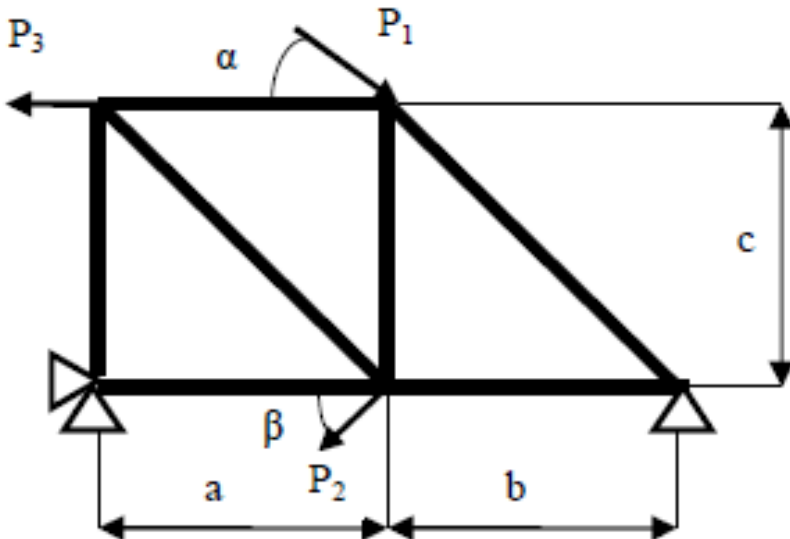


Рис. 6.1. Ферма № 1

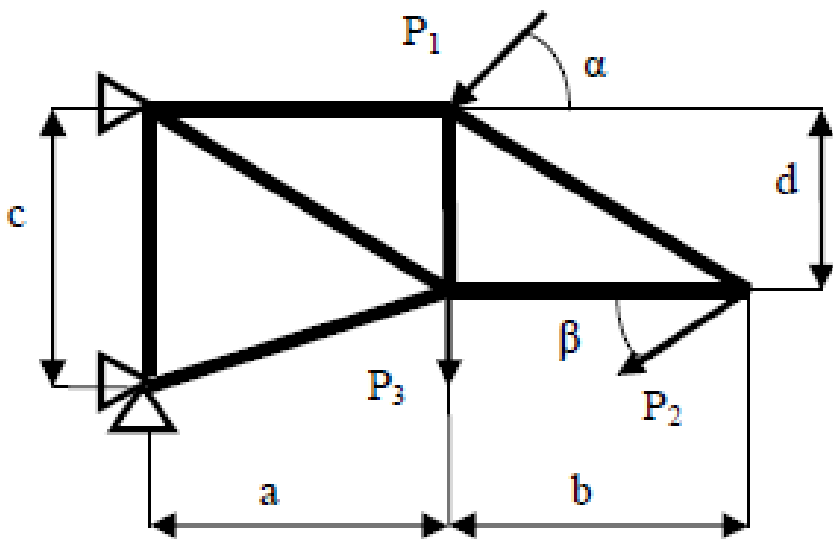


Рис. 6.2. Ферма № 2

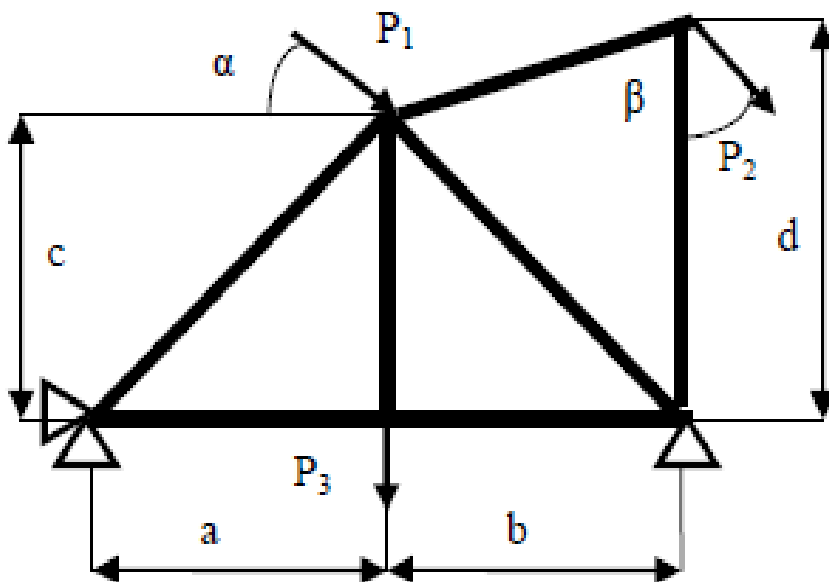


Рис. 6.3. Ферма № 3

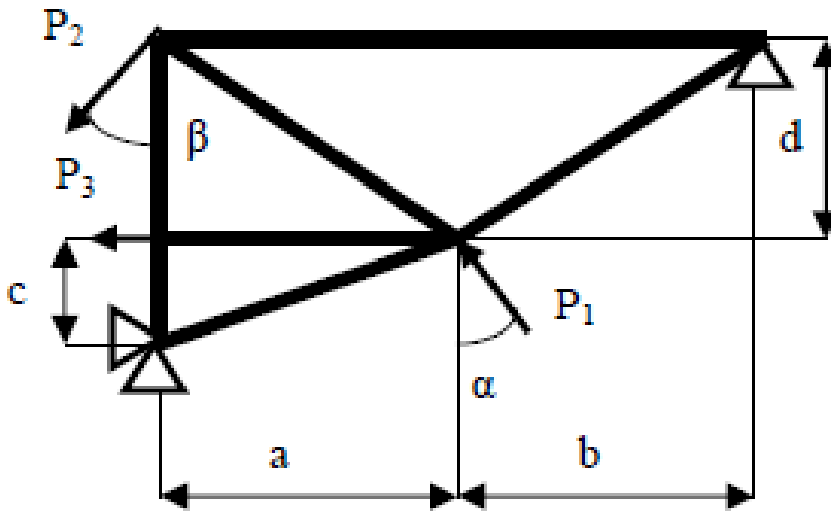


Рис. 6.4. Ферма № 4

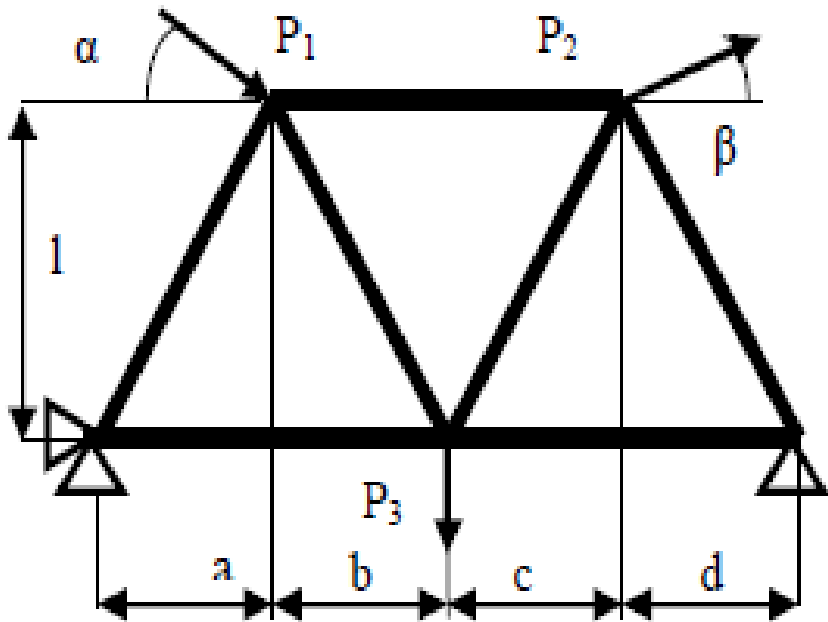


Рис. 6.5. Ферма № 5

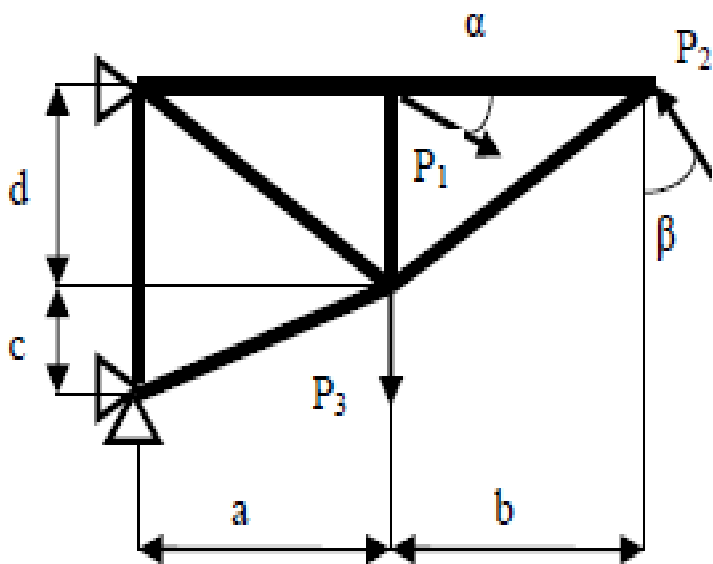


Рис. 6.6. Ферма № 6

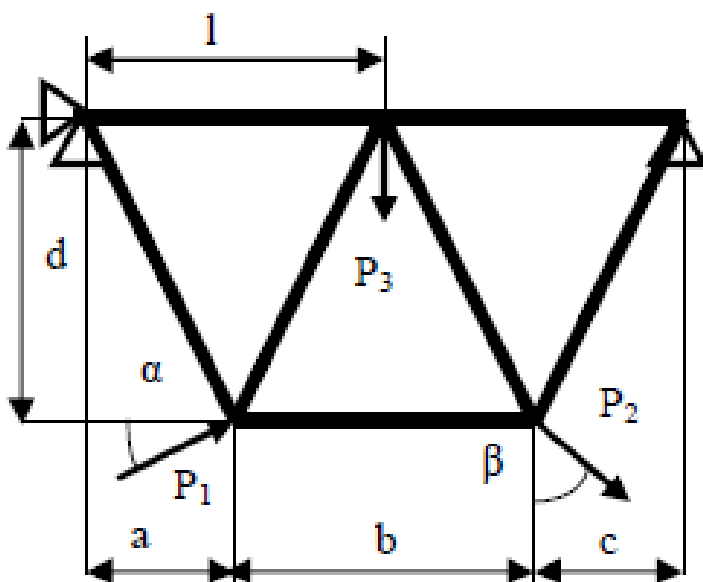


Рис. 6.7. Ферма № 7

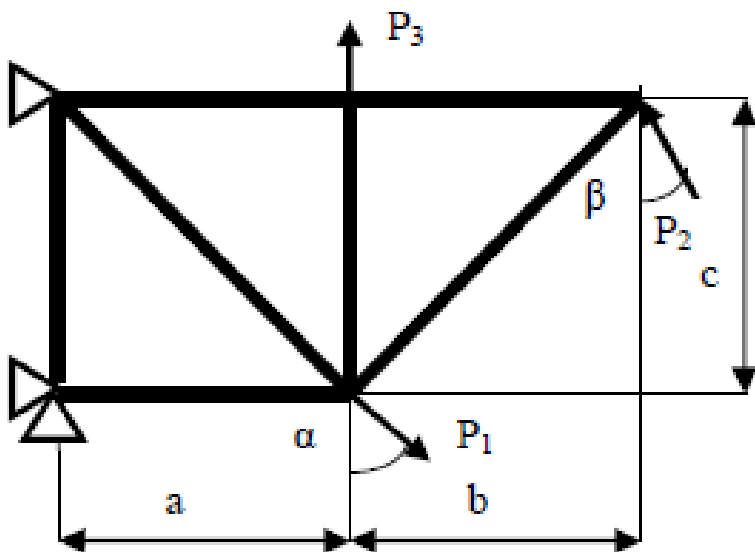


Рис. 6.8. Ферма № 8

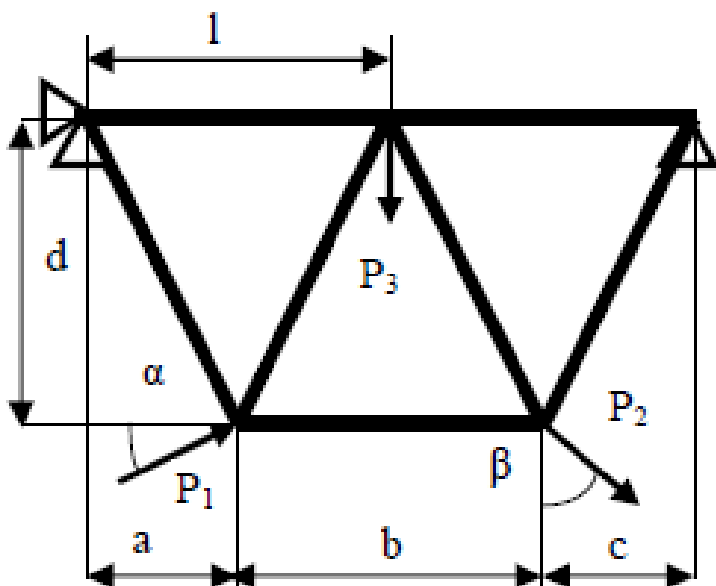


Рис. 6.9. Ферма № 9

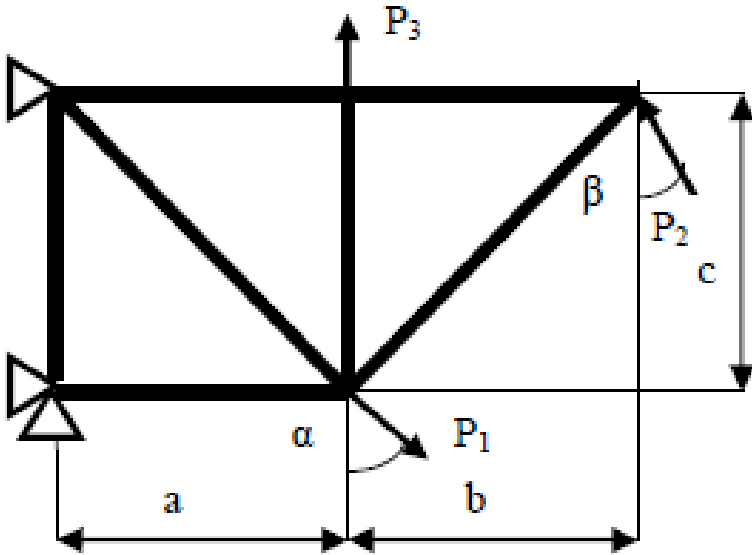


Рис. 6.10. Ферма № 10

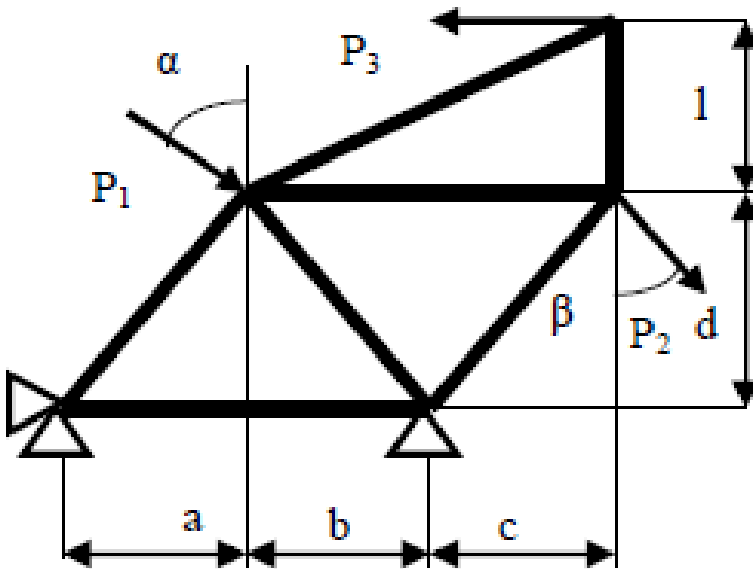


Рис. 6.11. Ферма № 11

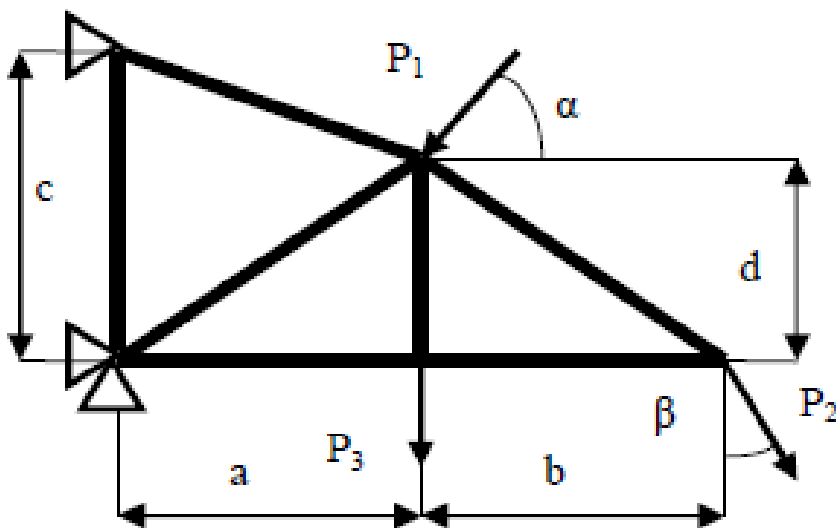


Рис. 6.12. Ферма № 12

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какая конструкция называется фермой?
2. Как нагружается и закрепляется ферма?
3. Какой конечный элемент используется для расчета ферм?
4. Какие деформации имеет ферменный конечный элемент?
5. Степени свободы узла фермы.
6. Векторы степеней свободы ферменных элементов и фермы.
7. Векторы узловых сил ферменных элементов.
8. Основные операции моделирования фермы в препроцессоре.
9. Основные операции расчета фермы в процессоре-решателе.
10. Основные операции анализа фермы в постпроцессоре.
11. Как определяется опасный стержень фермы.
12. Как строится командный файл для моделирования фермы?

ЛИТЕРАТУРА

1. Мещерский, И. В. Сборник задач по теоретической механике / И. В. Мещерский. – М.: Наука, 1986. – 448 с.
2. Наседкин, А. В. Конечно-элементное моделирование на основе ANSYS: Программы решения статических задач сопротивления материалов с вариантами индивидуальных заданий / А. В. Наседкин. – Ростов-на-Дону: УПЛ РГУ, 1998. – 44 с.
3. Расчет с помощью пакета ANSYS ферм. Расч. граф. раб № 1 по курсу «Численные методы расчета в инженерных задачах» / сост. В. Г. Фокин. – Самара: Самар. гос. техн. ун-т, 2014. – 14 с.
4. Клебанов, Я. М. Современные методы компьютерного моделирования процессов деформирования конструкций : учеб. пособие / Я. М. Клебанов, В. Г. Фокин, А. Н. Давыдов. – Самара: Самар. гос. техн. ун-т, 2004. – 100 с.
5. Конохов, А. В. Основы анализа конструкций в ANSYS / А. В. Конохов. – Казань: Казанский государственный университет, 2001. – 142 с.
6. Каплун, А. Б. ANSYS в руках инженера : практическое руководство / А. Б. Каплун, Е. М. Морозов, М. А. Олферьева. – М.: Едиториал УРСС, 2003. – 272 с.
7. Напрасников, В. В. Построение конечно-элементной модели на основе языка APDL : учебно-методическое пособие / В. В. Напрасников [и др.]. – Минск: БНТУ, 2009. – 51 с.
8. Напрасников, В. В. Создание конечно-элементной модели для расчета контейнера в процессе прессования порошковой заготовки : лабораторный практикум / В. В. Напрасников [и др.]. – Минск: БНТУ, 2008. – 89 с.
9. Напрасников, В. В. Конечно-элементное моделирование в ANSYS в режиме удаленного доступа к суперкомпьютеру «СКИФ» : учебно-методическое пособие / В. В. Напрасников, А. В. Бородуля, В. А. Кочуров. – Минск: БНТУ, 2008. – 65 с.
10. Напрасников, В. В. Моделирование колебаний рамной конструкции на основе метода конечных элементов : учебно-методическое пособие / В. В. Напрасников [и др.]. – Минск: БНТУ, 2010. – 43 с.
11. Напрасников, В. В. Создание 3D конечно-элементной модели в среде ANSYS : учебно-методическое пособие / В. В. Напрасников [и др.]. – Минск: БНТУ, 2010. – 37 с.

12. Напрасников, В. В. Расчет температурных полей на основе конечно-элементного моделирования : учебно-методическое пособие / В. В. Напрасников [и др.]. – Минск: БНТУ, 2011. – 22 с.

13. Напрасников, В. В. Оптимизационные расчеты на основе командного файла в ANSYS : учебно-методическое пособие / В. В. Напрасников, Ю. В. Напрасникова. – Минск: БНТУ, 2014. – 20 с.

14. Напрасников, В. В. Особенности расчета конструкций с шарнирами в ANSYS / В. В. Напрасников, Ю. В. Напрасникова, А. Н. Соловьев. – Минск: БНТУ, 2013. – 37 с.

15. Волков, В. М. Алгоритмическая реализация численных методов : учебно-методический комплекс / В. М. Волков, И. Л. Ковалева, В. В. Напрасников. – Минск: БНТУ, 2018. – 63 с.

16. Главная – CAE-services [Электронный ресурс]. – Режим доступа: <http://www.cadfem.ru/>. – Дата доступа: 24.05.2018.

17. Центр компьютерного инжиниринга СПбПУ [Электронный ресурс]. – Режим доступа: <http://www.cae.ru/>. – Дата доступа: 24.05.2018.

18. ANSYS. Basic Analysis Procedures Guide. Rel. 5.3. / ANSYS Inc. Houston [Электронный ресурс]. – 1994. – Режим доступа: <https://ru.scribd.com/document/144635033/Ansys-Basic-Analysis-Guide>. – Дата доступа: 25.05.2018.

19. ANSYS Mechanical APDL Element Reference [Электронный ресурс]. – Режим доступа: https://www.mm.bme.hu/~gyebro/files/vem/ansys_14_element_reference.pdf. – Дата доступа: 24.05.2018.

20. Mechanical APDL Element Reference [Электронный ресурс]. – Режим доступа: https://www.mm.bme.hu/~gyebro/files/ans_help_v182/ans_elem/Hlp_E_ElemTOC.html. – Дата доступа: 24.05.2018.

21. ANSYS. Theory Reference. Rel. 5.3. Ed. P. Kothnke / ANSYS Inc. Houston [Электронный ресурс]. – 1994. – Режим доступа: <http://research.me.udel.edu/~lwang/teaching/MEex81/ansys56manual.pdf>. – Дата доступа: 24.05.2018.

22. ANSYS Mechanical APDL Verification Manual. Release 15.0 / ANSYS, Inc Houston [Электронный ресурс]. – 2013. – Режим доступа: <http://docshare01.docshare.tips/files/22829/228290690.pdf>. – Дата доступа: 24.05.2018.

23. ANSYS Mechanical APDL Theory Reference, Release 15.0 / ANSYS, Inc Houston [Электронный ресурс]. – 2013. – Режим доступа: <http://lme.e.univ-evry.fr/~hello/ENS/CS93/ANSYS/ANSYS%20Mechanical%20APDL%20Theory%20Reference.pdf>. – Дата доступа: 24.05.2018.

24. ANSYS 10.0 Advanced Analysis Techniques Guide ANSYS, Inc Houston [Электронный ресурс]. – 2005. – Режим доступа: <https://imechanica.org/files/AdvacedAnsys.pdf>. – Дата доступа 24.05.2018.

Учебное издание

НАПРАСНИКОВ Владимир Владимирович
БОРОДУЛЯ Алексей Валентинович
КОВАЛЕВА Ирина Львовна и др.

**КОМПЬЮТЕРНОЕ
КОНЕЧНО-ЭЛЕМЕНТНОЕ
МОДЕЛИРОВАНИЕ**

Пособие для студентов специальности
1-40 05 01 «Информационные системы и технологии
(по направлениям)» направления специальности
1-40 05 01-01 «Информационные системы и технологии
(в проектировании и производстве)»

В 2 частях

Часть 1

Редактор *А. С. Мокрушников*
Компьютерная верстка *Е. А. Беспанской*

Подписано в печать 01.09.2021. Формат 60×84 ¹/₁₆. Бумага офсетная. Ризография.
Усл. печ. л. 4,88. Уч.-изд. л. 3,82. Тираж 100. Заказ 708.

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.