



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДФУ)

ИНЖЕНЕРНАЯ ШКОЛА

Кафедра механики и математического моделирования

Мокроусов Игорь Михайлович

Моделирование сплошной среды ферменными конструкциями на примере цилиндрической оболочки при действии всестороннего внешнего давления

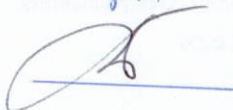
ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

по направлению подготовки 15.03.03 Прикладная механика,
профиль «Математическое и компьютерное моделирование механических систем и процессов»

Владивосток

2018

Сведения, содержащая
гос. тайну, нет.
Эксперт

 А.А. Бочарова

Автор работы Мокрухов А.М.
(подпись)
« 4 » июля 2018 г.

Консультант(ы)*
Всудов С.В. Субботкин В.В.
(подпись) (И.О. Фамилия)

« 5 » июля 2018 г.

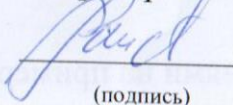
Руководитель ВКР Ратников А.А.
(подпись) (И.О. Фамилия)
старший преподаватель
(должность, учёное звание)

« 03 » июля 2018 г.

Защищена в ГЭК с оценкой

отлично


Секретарь ГЭК

 А.А. Ратников
(подпись)

« 04 » июля 2018 г.

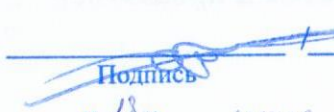
«Допустить к защите»

Заведующий кафедрой, доцент

 А.А. Бочарова
(подпись)

« 03 » 07 2018 г.

УТВЕРЖДАЮ
Директор Инженерной школы

 Беккер А.Г.
Подпись / Ф.И.О.
« 13 » июля 2018 г.

В материалах данной выпускной квалификационной работы не содержатся сведения, составляющие государственную тайну, и сведения, подлежащие экспортному контролю.

Уполномоченный по экспортному контролю

_____/_____/ «__» _____ 201 г.
Ф.И.О. Подпись

Оглавление

Введение	2
Глава 1. Ферма, как модель сплошной среды, методы и способы её расчёта.	3
1.1. Понятие, элементы, свойства и применение фермы.	3
1.2. Модель сплошной среды.....	5
1.3. Цилиндрическая оболочка, описание, граничные условия.....	6
1.4. Методы расчёта	8
Глава 2. Расчёт ферменных конструкций с помощью программного комплекса.	14
2.1. Постановка задачи.....	14
2.2. Программный комплекс.....	15
2.3. Разработка программы	17
2.4. Моделирование ферменной конструкции.....	22
Заключение	25
Литература	26
Приложение	27

Введение

Потребности современного строительства в таких аспектах, как свай большой длины для фундамента высотных домов всё больше и больше вызывают широкое использование цилиндрических оболочек, одним из главных достоинств которых является высокая прочность и сопротивление сжатию под большим давлением. В связи с этим предъявляются повышенные требования к качеству проектирования подобных конструкций, основанного на возможности предсказания их поведения при воздействии всестороннего внешнего давления.

Объектом изучения в данной работе является моделирование сплошной среды ферменными конструкциями.

Предметом изучения является изучение применимости замены сплошной среды ферменной конструкцией в криволинейных координатах.

Цель заключается в расчёте ферменной конструкции на основе цилиндрической оболочки при заданной нагрузке с последующим определением способа моделирования, позволяющего передать свойства реального материала и обеспечить точность решения, с использованием программы собственной разработки, а также программного комплекса для расчёта ферменных конструкций.

Для выполнения данной цели следует:

- Ввести модель сплошной среды для расчёта ферменных конструкций.
- Проанализировать методы расчета фермы на основе цилиндрической оболочки.
- Разработать программу для подготовки входного файла.
- Проанализировать файл расчёта программного комплекса с последующей визуализацией изменений цилиндрической оболочки под действием внешнего всестороннего давления.

Глава 1. Ферма, как модель сплошной среды, методы и способы её расчёта.

В данной работе рассматриваются ферменные конструкции, которые подвержены действию всестороннего внешнего давления. В современном мире фермы широко используются в строительстве.

1.1. Понятие, элементы, свойства и применение фермы

Ферма – это стержневая система, остающаяся неизменяемой геометрически при условной замене жестких узлов шарнирами.

Фермы работают на изгиб, а конструкция фермы состоит из нескольких стержней, которые соединяются в жестких узлах.

В стержнях фермы при действии нагрузок, приложенных в узлах, возникают продольные усилия сжатия или растяжения.

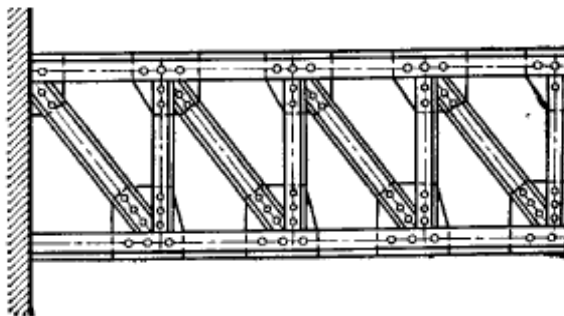


Рис.1. Пример фермы

Фермы классифицируются на плоские и пространственные. Плоские фермы отличаются от пространственных тем, что оси стержней плоской фермы лежат в одной плоскости, а у пространственной – нет. Расчет пространственной фермы во многих случаях можно свести к расчету нескольких плоских ферм.

Расстояние между осями опор фермы называется пролётом. Стержни, которые расположены по внешнему контуру фермы, называются поясными и образуют пояса. Стержни, соединяющие пояса, образуют решетку фермы и,

если стержни вертикальные – называются стойками, а если наклонные – раскосами. Расстояние между соседними узлами любого пояса фермы называется панелью.

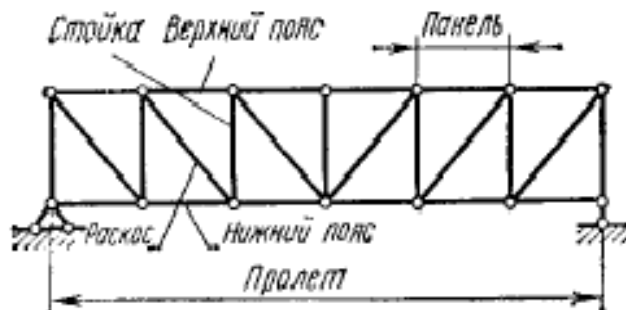


Рис. 2. Элементы фермы

Классификацию ферм можно провести по нескольким признакам:

- Характер очертания
- Тип решетки
- Тип опирания фермы
- Назначение фермы.

По характеру очертания различают фермы с параллельными поясами, с полигональным расположением поясов и фермы треугольного очертания.

По типу решетки фермы бывают: с треугольной решеткой, с раскосной решеткой, многорешетчатые фермы и др.

По типу опирания фермы бывают закреплёнными с обоих концов – балочными и закреплёнными с одного конца – консольными.

В зависимости от назначения фермы бывают стропильные, крановые, башенные и мостовые.

Фермы применяют при самых разнообразных нагрузках и в разных аспектах нашей жизни: в зависимости от назначения им придают самые разнообразные конструктивные формы – от легких прутковых конструкций до тяжелых ферм, стержни которых могут складываться из нескольких элементов крупных профилей или листов. Наибольшую популярность имеют разрезные

балочные фермы из-за простоты в изготовлении и монтаже. Неразрезные и консольные системы ферм рационально использовать при большой собственной массе конструкции, так как в этом случае они могут значительно сэкономить материал. Кроме того, неразрезные фермы можно применять опираясь на требования эксплуатации из-за того, что обладают большей жесткостью при меньшей высоте.

Фермы широко используются в современном строительстве, в основном для перекрытия больших пролётов для уменьшения расхода материалов и облегчения конструкций. Например, в строительных большепролётных конструкциях таких, как мосты, стропильные системы промышленных зданий, спортивные сооружения, а также небольшие лёгкие строительные и декоративные конструкции: павильоны, сценические конструкции, тенты и подиумы.

Также с точки зрения сопротивления материалов фермами являются: фюзеляж самолёта, корпус корабля, несущий кузов автомобиля, кроме кабриолетов, работающих, как балка, кузов автобуса, тепловоза, вагонная рама со шпренгелем.

1.2. Модель сплошной среды.

Под моделью сплошной среды понимают такую гипотетическую среду, в которой учтены только некоторые из физических свойств, существенные для определенного круга явлений и технических задач. Другие малосущественные свойства среды в модели игнорируются.

Основной гипотезой, на которой базируется сопротивление материалов, является гипотеза непрерывности (сплошности) материала твердого тела, согласно которой тело рассматривается как сплошная среда. Предполагаем также, что твердое тело изотропно и однородно, т. е. механические свойства во

всех направлениях одинаковы и не меняются при переходе от одной точки тела к другой.

В механике деформируемого твёрдого тела часто записывают уравнение неразрывности в виде связи между начальной и конечной плотностью материальной единицы. В случае малых деформаций уравнение имеет вид:

$$\rho = \rho_0(1 - \operatorname{div} w), \quad (1)$$

где ρ_0 – начальная плотность, ρ – конечная плотность, а w – вектор перемещений.

Данное уравнение неразрывности имеет универсальный характер и может применяться для любой сплошной среды.

1.3. Цилиндрическая оболочка, описание, граничные условия.

При расчёте на прочность и жёсткость конструкция представляет собой схему. В этом случае элементы конструкции представляем в виде бруса или оболочки. Оболочка – твёрдое тело, ограниченное двумя криволинейными поверхностями, расстояние между которыми мало по сравнению с двумя другими размерами. Поверхность, делящая пополам толщину оболочки – это срединная поверхность. В зависимости от очертания оболочки различают цилиндрическую оболочку сечением круговой формы, эллиптической, а также тороидальные, конические и т.д. (рис.3).

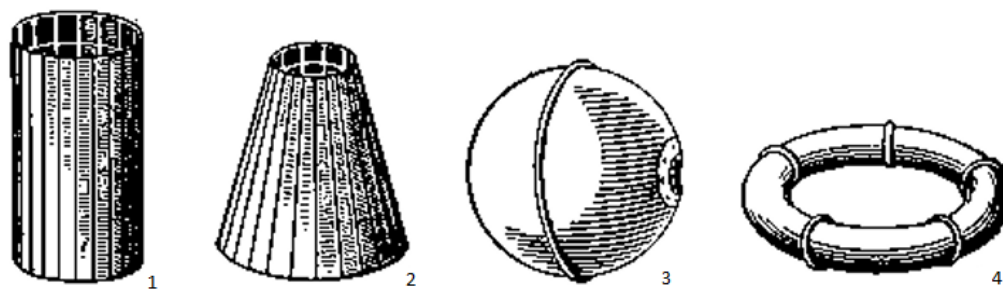


Рис. 3. Оболочки различной формы: 1 — цилиндрическая оболочка кругового сечения, 2 — коническая оболочка, 3 — сферическая оболочка, 4 — тороидальная оболочка.

Оболочки классифицируются также по полной кривизне поверхности – гауссовой кривизне: положительная – сферические, эллипсоидальные; нулевой – цилиндрические, конические; отрицательная – гиперболические параболоиды.

Оболочки могут быть постоянной и переменной толщины. Они подразделяются на однослойные, двухслойные и многослойные. По материалу оболочки классифицируются на изотропными и анизотропными. Выполняются оболочки из любых строительных материалов.

При воздействии внешних нагрузок в оболочке возникают внутренние усилия, равномерно распределённые по толщине, такие усилия называются – мембранными напряжениями, или напряжения в срединной поверхности. Также при внешних нагрузках появляются усилия изгиба, образующие в сечениях оболочки изгибающие, крутящие моменты и поперечные силы. Благодаря наличию мембранных усилий оболочки сочетают значительную прочность и жёсткость со сравнительно малым весом, в отличии от пластинок. Если напряжения изгиба при расчёте можно пренебречь, то оболочка называется безмоментной. Наличие моментов характерно для участков, примыкающих к краям. Если напряжения лежат в пределах пропорциональности для материала, тогда методы расчёта основываются на зависимостях теории упругости.

Чаще всего для тонких оболочек применяют гипотезу Кирхгофа — Лява, по которой любое прямое волокно, перпендикулярное к срединной поверхности до деформации, остаётся прямым и перпендикулярным к срединной поверхности после деформации, также его длина остаётся неизменной. Считается, что нормальными напряжениями в направлении, перпендикулярном к срединной поверхности, можно пренебречь по сравнению с основными напряжениями. При этом трёхмерная задача теории упругости переходит в двумерную, решение которой сводится к интегрированию системы ДУ в

частных производных высокого порядка при краевых условиях, определяемых характером сопряжения оболочки с другими частями конструкции.

В статическом расчёте на прочность и жёсткость должны быть определены напряжения, деформации и перемещения различных точек в зависимости от заданной нагрузки. Как правило, в расчётах на прочность перемещения вдоль нормали к срединной поверхности оболочки могут считаться малыми по сравнению с толщиной, следовательно, соотношения между перемещениями и деформациями являются линейными. Значит линейными будут и основные дифференциальные уравнения.

Важным для оболочек является расчёт на устойчивость. Особенность тонкостенных оболочек – потеря устойчивости хлопком, или прощёлкиванием, выражается в резком переходе от одного устойчивого равновесного состояния к другому. Этот переход наступает при различных нагрузках, в зависимости от исходных несовершенств формы оболочки, начальных напряжений и т.д. В случае прощёлкивания прогибы оказываются соизмеримыми с толщиной оболочки. Анализ поведения должен основываться при этом на нелинейных уравнениях.

1.4. Методы расчёта

Для расчёта пространственных ферм существует большое количество способов и методов различной сложности и для разных ситуаций, от классических методов вроде вырезания узлов и метода сечений, до специализированных программ расчёта. Рассмотрим некоторые из них.

Метод вырезания узлов

Метод заключается в вырезании узлов фермы и рассмотрении их равновесия. Его можно реализовать как аналитическим методом, так и

графическим. Аналитический метод более точный, т.к. не имеет, например, погрешности линейки. Вырезать узлы фермы следует так, чтобы число неизвестных усилий стержней вырезанного узла было не больше двух. Первый вырезанный узел не должен содержать более двух стержней. Усилия в отдельных стержнях фермы могут быть нулевыми, для определения таких стержней существует 3 леммы [6].

Лемма 1. Если в незагруженном узле плоской фермы сходятся два стержня, то усилия в этих стержнях равны нулю (рис.4).

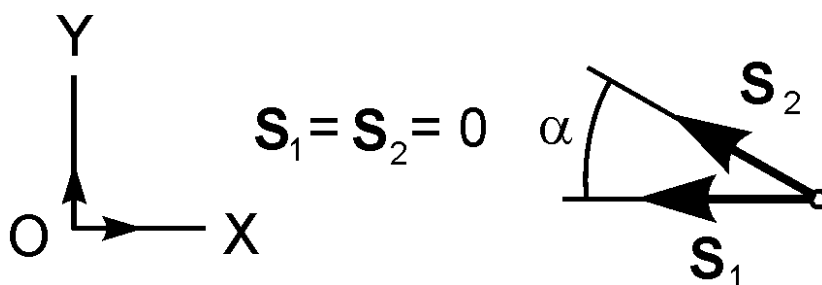


Рис.4. Наглядное изображение леммы 1.

Лемма 2. Если в незагруженном узле плоской фермы сходятся три стержня, из которых два расположены на одной прямой, то усилие в третьем стержне равно нулю. Усилия в первых двух стержнях равны между собой (рис. 5).

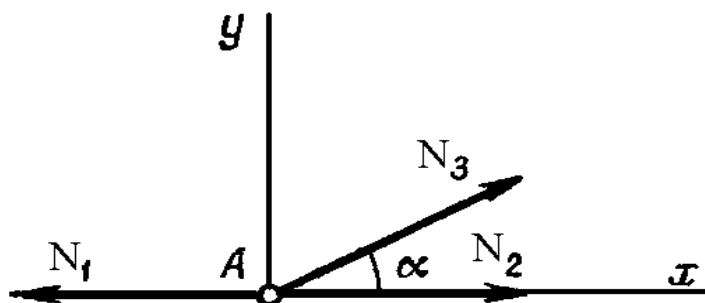


Рис. 5. Наглядное изображение леммы 2.

Лемма 3. Если в узле плоской фермы сходятся два стержня и к узлу приложена внешняя сила, линия действия которой совпадает с осью одного из стержней, то усилие в этом стержне равно по модулю приложенной силе, а усилие в другом стержне равно нулю (рис.6).

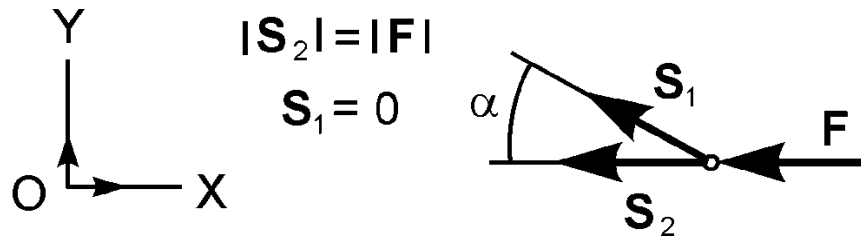


Рис. 6. Наглядное изображение леммы 3.

Способ сквозных сечений (Способ Риттера)

В этом способе ферма рассекается на две части так, чтобы кроме действующих на рассматриваемую часть фермы активных стержней и реакций связей, прикладывались усилия в рассеченных стержнях. Все стержни предполагаются растянутыми и число рассеченных стержней, усилия в которых неизвестны, не должны быть больше трёх. В основном составляются уравнения моментов относительно точек Риттера – точек пересечения линий действия усилий в рассеченных стержнях.

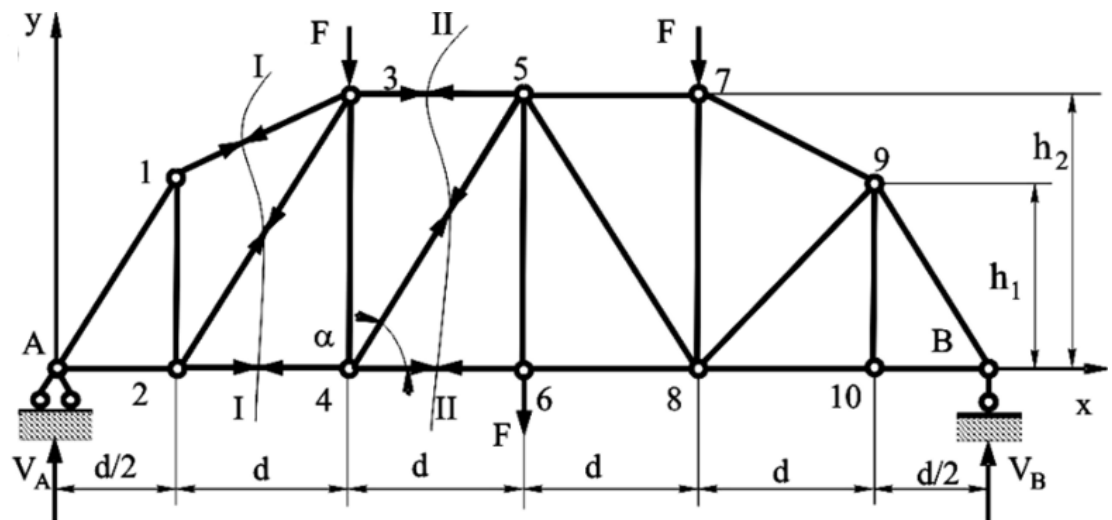


Рис. 7. Пример фермы, рассечённой методом Риттера.

Современный метод расчёта ферм

Современные программы позволяют рассчитывать фермы с помощью численных методов, например, таких как метод конечных элементов, где ферму можно анализировать как конечно – элементную модель, стержни которой являются одномерными конечными элементами, испытывающими только растяжение – сжатие. Шарниры представляют собой узлы конечно – элементной модели. За степень свободы берутся узловые перемещения, которые считаются малыми.

Конечный элемент фермы представлен на рис.8. Он связан с местной системой координат \bar{x}, \bar{y} . Общая для всей фермы система координат x, y параллельно перенесена в узел i .

В местной системе координат векторы узловых перемещений и узловых сил элемента имеют вид: $\{\bar{U}\}_e = \begin{Bmatrix} \bar{U}_i \\ \bar{U}_j \end{Bmatrix}_e, \{\bar{F}\}_e = \begin{Bmatrix} \bar{F}_i \\ \bar{F}_j \end{Bmatrix}_e$.

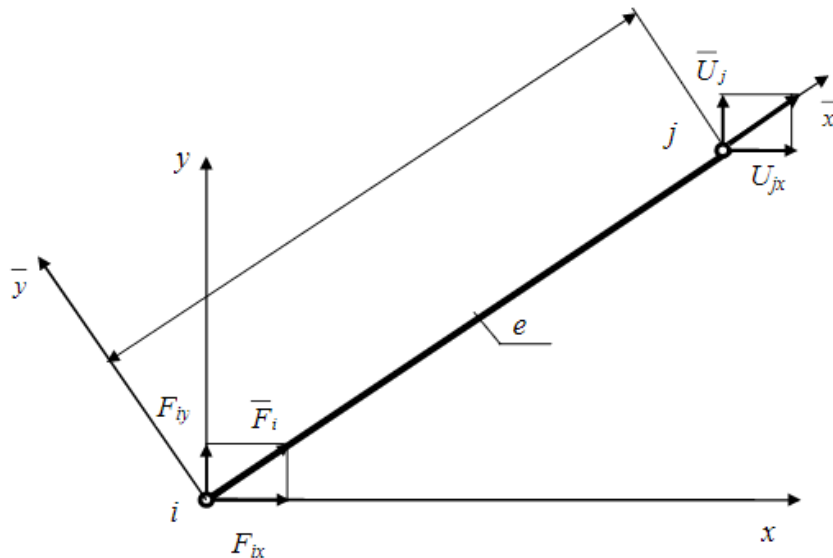


Рис. 8. Ферменный конечный элемент.

В теории МКЭ устанавливается связь между векторами узловых перемещений и узловых сил элемента в виде соотношения матриц:

$$[K]_e \{U\}_e = \{F\}_e, \quad (2)$$

где $[K]_e$ – матрица жесткости элемента.

С учётом предыдущего выражения условия равновесия узлов фермы разрешающие систему уравнений МКЭ:

$$[K]\{U\} = \{P\}, \quad (3)$$

где $[K]$ – глобальная матрица жесткости конечно – элементной модели, которая состоит из матриц жесткости элементов.

Для единственности решения уравнений (3) необходимо учесть граничные условия по перемещениям и закрепления. Конструкция должна закрепляться так, чтобы было невозможным перемещение её как твёрдого тела. В результате решения системы уравнений (3) определяются узловые перемещения $\{U\}$, через которые, исходя из формул теории упругости, вычисляются все другие искомые величины.

Метод, используемый в программном комплексе

Рассмотрим пространственную ферму, состоящую на S стержней и K подвижных узлов. Пронумеруем все узлы фермы, включая неподвижные, и отнесём ферму к прямоугольной системе координат XOY .

Для каждого подвижного j -ого узла пространственной фермы можно записать три независимых уравнения статики:

$$\begin{aligned} \sum_i N_{ji} l_{ji} + F_{jx} &= 0 \\ \sum_i N_{ji} m_{ji} + F_{jy} &= 0 \\ \sum_i N_{ji} n_{ji} + F_{jz} &= 0, \end{aligned} \quad (4)$$

где n_{ji} – косинус угла между вектором N_{ji} и осью OZ ; F_{jz} – проекция внешней силы, действующей на узел, на ось OZ . Правило знаков для F_{jz} такое же, как для F_{jx} и F_{jy} (обычное как для проекции).

Общее количество уравнений (4) для всей фермы очевидно равно $3K$.

Исключая уравнения $-N_{ji} + \sum F_{ji} + N_{ij} = 0$, которые при действии внешних сил только на углы фермы принимают вид: $N_{ji} = N_{ij}$, можно считать, что число неизвестных усилий равно числу стержней, то есть S . Следовательно, для степени статической неопределимости пространственной фермы имеет формулу:

$$r = S - 3K. \quad (5)$$

Также как и для плоских ферм выделим в уравнениях (4) усилия, действующие на узел от жестких стержней. Тогда для j -ого узла уравнения равновесия можно записать в виде:

$$\sum_i N_{ji} l_{ji} + \sum_k N_{jk} l_{jk} + F_{jx} = 0$$

$$\sum_i N_{ji} m_{ji} + \sum_k N_{jk} m_{jk} + F_{jy} = 0 \quad (6)$$

$$\sum_i N_{ji} n_{ji} + \sum_k N_{jk} n_{jk} + F_{jz} = 0,$$

где N_{jx} – усилие в жестком стержне.

Если узел плоский, то третье уравнение обращается в тождественное $0 \equiv 0$ и уравнения (6) переходят в уравнения:

$$\sum_i N_{ji} l_{ji} + \sum_k N_{jk} l_{jk} + F_{jx} = 0$$

$$\sum_i N_{ji} m_{ji} + \sum_k N_{jk} m_{jk} + F_{jy} = 0.$$

Глава 2. Расчёт ферменных конструкций с помощью программного комплекса.

2.1. Постановка задачи.

Рассмотрим цилиндрическую оболочку конечной длины h , внешним радиусом $R1$ и внутренним радиусом $R2$, закреплённый с одной стороны жесткой заделкой, а с другой свободный. На цилиндрическую оболочку действует всестороннее внешнее давление F .

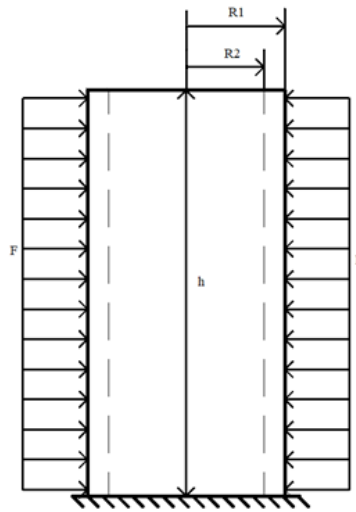


Рис. 9. Полый цилиндр.

Требуется смоделировать ферменную конструкцию, максимально приближенную по свойствам к реальному материалу цилиндрической оболочки. Конечным элементом, которыми будем моделировать ферменную конструкцию, как цилиндрическую оболочку выберем ферму вида, представленного на рис.10, т.к. требуется приблизить свойства реального материала, а именно коэффициент Пуассона, к свойствам ферменной конструкции.

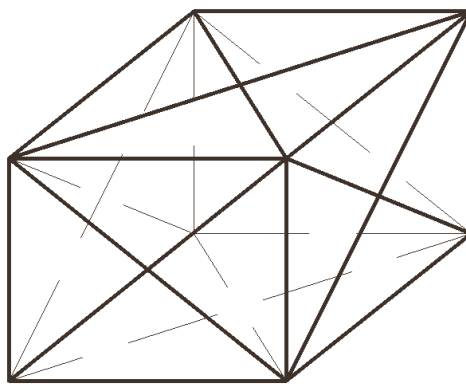


Рис 10. Конечный пространственный элемент

2.2. Программный комплекс

Расчёт пространственной фермы цилиндрической оболочки производится с помощью программного комплекса расчета статически неопределимых пространственных ферм. Данный программный комплекс – это реализация в рамках персональных ЭВМ рассматриваемой методики. В основу реализации программного обеспечения положен модульный принцип. Каждый модуль комплекса представлен своей компонентой – подзадачей.

Доступ к компонентам осуществляется посредством «меню». Для запуска данного программного комплекса (рис. 11) потребовалась установка эмулятора DOS программ. Выбор компонент в окне эмулятора производится с помощью стрелок на клавиатуре. Для выбора файла начальных данных для программного комплекса требуется переместить файл в корневую папку программного комплекса, выбрать компоненту «Имя файл данных фермы» и ввести название файла начальных данных, например «path0.D».

Далее следует выбрать компоненту «Проектировочный расчёт фермы» для выполнения расчётов и создания файла расчета данных в корневой папке программного комплекса. После приведённых действий сформируется файл с расчётом заданной ферменной конструкции, в котором будет полная

информация об усилиях в стержнях, напряжениях в стержнях, длинах стержней и об отклонениях стержней от длины, а также с информацией об изменении координат узлов.

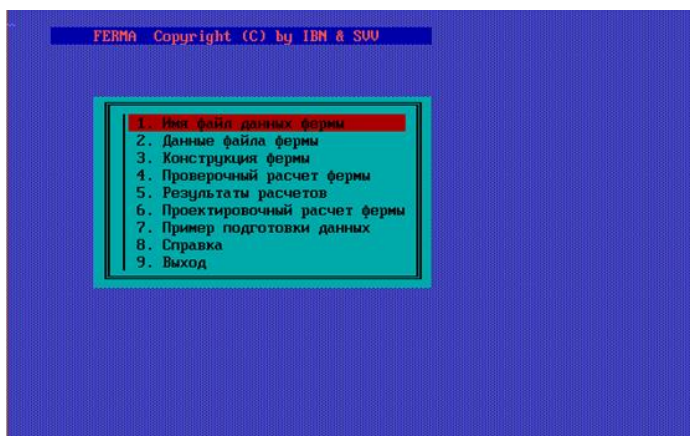


Рис. 11. Главное меню программного комплекса

Также в программном комплексе существует возможность графического представления ферменной конструкции для проверки результатов нумерации узлов, соединения этих узлов стержнями и верное расположение сил в узлах. Для запуска графического представления требуется выбрать компоненту «Конструкция фермы» (рис. 12).

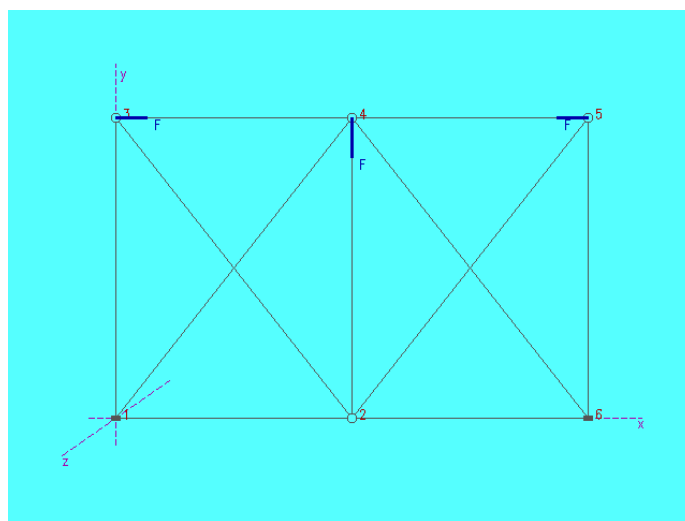


Рис.12. Графическое представление ферменной конструкции

2.3. Разработка программы

Программный комплекс расчёта статически неопределимых пространственных ферм имеет большой потенциал в расчете ферменных конструкций любой сложности и нагромождённости, кроме того, имеет предельно понятный и лаконичный интерфейс.

Единственное, что затрудняет расчет ферменных конструкций большого объёма – это кропотливость и повышенное внимание при составлении входного файла для программного комплекса.

Именно эту проблему требуется решить путём разработки программы для формирования входного файла. Язык написания программы выберем C#, т.к. данный язык отлично подходит для написания приложений ОС Windows, а также имеет большое количество стандартных инструментов для создания пользовательского интерфейса.

Для начала выберем разрядность x86, т.к. она поддерживается и в 32битных, и в 64битных ОС. Следом создадим пользовательский интерфейс, с которым будем работать в последствие (Рис. 13). В нём должны располагаться такие элементы, как: форма для задания количества точек по окружности цилиндра, форма для задания количества точек по длине цилиндра, форма для задания количества точек по стенке цилиндра, форма для задания внешнего радиуса цилиндрической оболочки, форма для задания модуля Юнга для стержней, форма для задания внутреннего радиуса цилиндрической оболочки, форма для задания длины цилиндрической оболочки, форма для задания всестороннего давления, действующего на цилиндрическую оболочку.

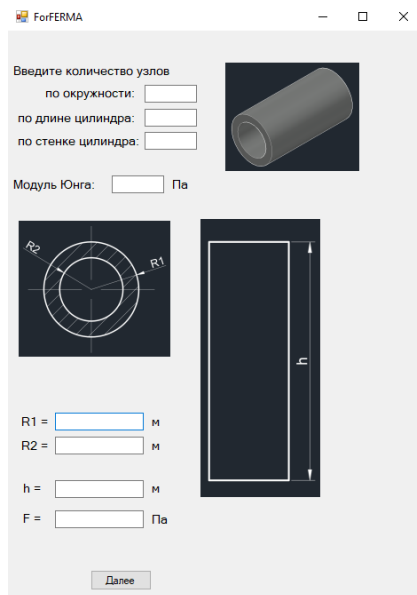


Рис. 13. Главный экран.

При заполнении всех требуемых полей реализован переход на следующий экран по средством нажатия на кнопку «Далее». При этом сразу запускается процесс заполнения входного файла.

Опираясь на структуру входного файла, для начала программа заносит в файл информацию о количестве узлов, стержней, а также количестве сил, действующих на ферменную конструкцию. Для расчёта количества узлов программа использует следующую формулу:

$$N_{\text{узлов}} = n_i n_j n_s, \quad (7)$$

где n_i – это количество узлов по окружности цилиндрической оболочки, n_j – количество узлов по длине цилиндрической оболочки, n_s – количество узлов по стенке цилиндрической оболочки.

Далее требуется рассчитать количество стержней в ферменной конструкции. Данное число высчитывается так:

```
for (int s = 0; s < num_n_s; ++s)
{
    for (int j = 0; j < num_n_h; ++j)
    {
        for (int i = 1; i <= num_n; ++i)
```

```

{
  --i;
  if (j != 0)
    num_rods += 1;
  if ((s + 1) < (num_n_s))
  {
    if (j != 0)
      num_rods += 3;
  }

  if ((j + 1) < (num_n_h))
  {
    num_rods += 2;
  }

  if ((j + 1) < (num_n_h))
    if ((s + 1) < (num_n_s))
    {
      num_rods += 2;
    } ...
}

```

В конце программа вычисляет количество сил, действующих на ферменную конструкцию цилиндрической оболочки. Так как нагрузка задана, как всесторонняя, то действовать она будет на все внешние узлы, кроме закреплённых, следовательно, количество сил равно

$$N_{\text{сил}} = n_i(n_j - 1). \quad (9)$$

Следующим шагом программа просчитывает координаты и нумерует каждый узел по порядку. Просчёт узлов по координатам осуществляется в основном посредством просчёта координат x и y :

$$\begin{cases} x = \cos(180l / 90n_iPI)(R1 - s(R1 - R2)) / (n_s - 1) + R1, \\ y = \sin(180l / 90n_iPI)(R1 - s(R1 - R2)) / (n_s - 1) + R1, \\ z = jh / (n_j - 1), \end{cases} \quad (10)$$

где l – это номер узла по окружности минус один, $R1$ – это внешний радиус цилиндрической оболочки, $R2$ – внутренний радиус, s – это номер узла

по стенке цилиндрической оболочки минус один, j – это номер узла по длине цилиндрической оболочки минус один и h – длина цилиндрической оболочки.

Для просчёта стержней, а именно номеров узлов, между которыми заключён стержень программа использует формулу, где i – это номер узла по окружности цилиндрической оболочки:

$$(i + n_i j + n_i n_j s). \quad (11)$$

В последнюю очередь программа просчитывает проекции сил, которые имитируют всестороннее давление F , действуя на незакреплённые узлы по осям x , y и z :

$$\begin{cases} x = F \cos(180l / 90n\pi), \\ y = F \sin(180l / 90n\pi), \\ z = j. \end{cases} \quad (12)$$

Экран загрузки (рис. 14), на котором происходит просчёт всей информации для входного файла представляет собой информативный стенд с полем динамического вывода координат узлов, полосой завершения процесса заполнения входного файла и кнопками управления процессом заполнения «Остановить» и «Очистить», а также с кнопкой возврата на Главный экран.

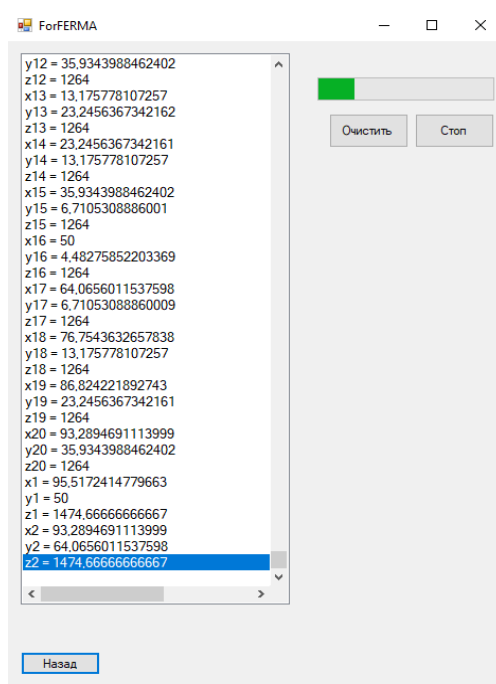


Рис. 14. Экран загрузки.

При нажатии на кнопку «Стоп» процесс заполнения входного файла будет немедленно остановлен без возможности возобновить процесс. Кнопка «Очистить» служит для полной очистки памяти ПК от информации для входного файла. По завершению процесса заполнения входного файла приложение проинформирует об этом надписью «Готово» на месте полосы загрузки, а также появится кнопка «Сохранить файл FERMA», при нажатии на которую приложение сохранит все данные в текстовый документ «path0.D» или далее по нумерации при найденном одноимённом файле в каталоге, где находится само приложение.

Для выполнения расчёта по полученному входному файлу в программном комплексе требуется переместить файл в корневую папку с программным комплексом, выбрать данный файл, вписав его название в компоненту «Имя файл данных фермы». Затем вписать нужное название файла расчёта в компоненту «Проектировочный расчёт фермы» для формирования файла расчёта для заданной ферменной конструкции цилиндрической оболочки.

2.4. Моделирование ферменной конструкции.

Для моделирования ферменной конструкции выберем цилиндрическую оболочку длины 40мм, внешним радиусом 10мм, внутренним радиусом 7мм на которую действует постоянное внешнее давление: 100 КПа. Модуль Юнга равен $2e^{11}$ МПа, а коэффициент Пуассона: 0,3.

Введём все данные в программу, а также укажем количество узлов по окружности цилиндрической оболочки: 10, по длине цилиндрической оболочки: 40 и по стенке: 3. Полученный входной файл, загрузим в программный комплекс и выполним Проектировочный расчёт. В итоге сформируется файл, который в последствие можно вывести графически, как изменилась ферменная конструкция после деформации (рис. 15).

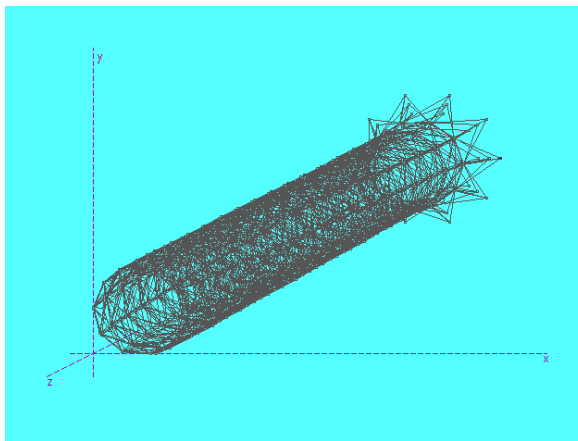


Рис 15. Ферменная конструкция после деформации.

Во входном файле найдём информацию о том, насколько узел, на свободном конце ферменной конструкции, отклонился по оси z от первоначального значения для обозначения изменения длины цилиндрической оболочки при действии всестороннего внешнего давления. Вычислим расстояние между 2-мя противоположными на окружности узлами на

свободном конце для определения изменения диаметра цилиндрической оболочки при действии всестороннего внешнего давления. В конце воспользуемся формулой для определения коэффициента Пуассона:

$$\mu = \left| \frac{\varepsilon_{\perp}}{\varepsilon} \right|. \quad (13)$$

Из файла результатов было выявлено, что отклонение точки z составило 9,018 мм, а расстояние между 2-мя противоположными на окружности узлами равно 11,242 мм, следовательно, диаметр изменился на 2,758 мм. Подставляем значения в формулу (13) и получаем коэффициент Пуассона, равный 0,306.

Осуществим расчёт идентичной цилиндрической оболочки в программном комплексе ANSYS Workbench 19.0 для проверки других величин, таких как максимальное напряжение и изменение радиуса стержня.

Максимальное изменение диаметра составило 2,5873 мм при 2,758 мм в файле расчётов программного комплекса:

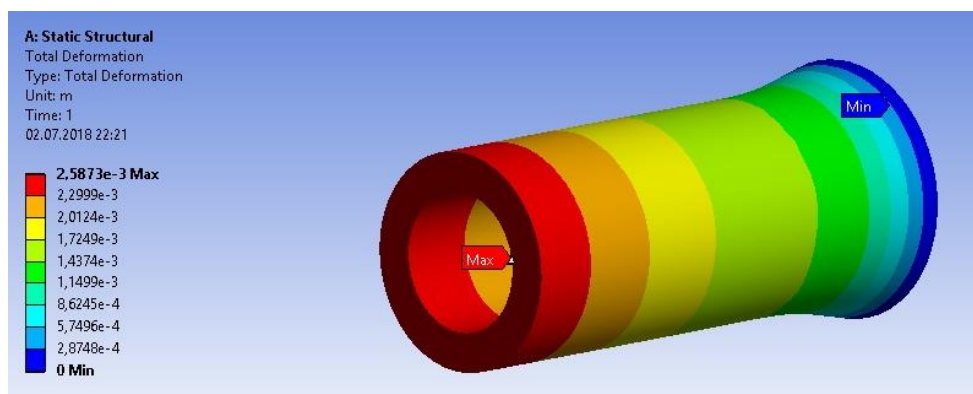


Рис 16. Максимальное изменение радиуса в ANSYS WB 19.0.

Максимальная нагрузка составила $4,1357e8$ Па при $4.5819e8$ Па в файле расчётов программного комплекса:

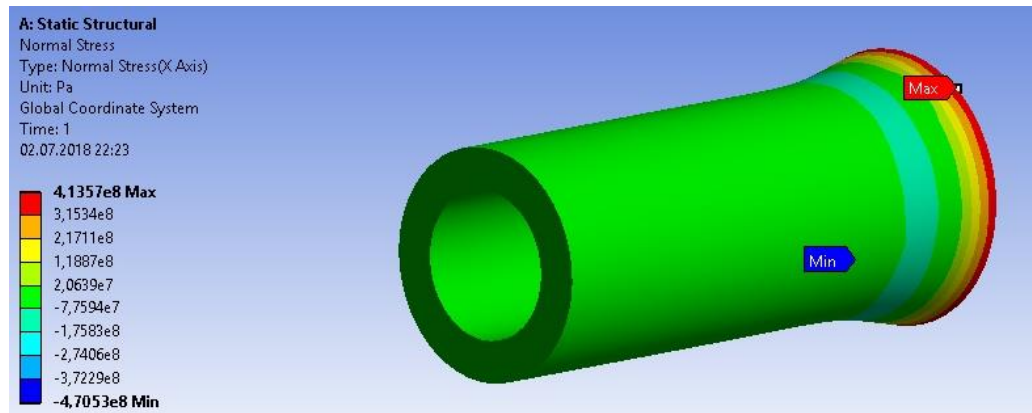


Рис 17. Максимальная нагрузка в ANSYS WB 19.0.

Из приведённых сравнений можно сделать вывод, что цилиндрическую оболочку возможно представить в виде ферменной конструкции. Чем больше узлов в ферменной конструкции, тем ближе к реальному материалу будут свойства ферменной конструкции.

Заключение

В данной работе были рассмотрены такие понятия как ферма, оболочка, модель сплошной среды. Изучены различные способы и методы расчёта ферм и ферменных конструкций, от базовых – аналитических и графических, до современных с использованием программных комплексов.

Была сформулирована постановка задачи, связанная с реальным объектом и предоставленными данными. Сформулирована проблема программного комплекса расчёта пространственных ферменных конструкций и описан способ её решения. Была описана работа программного комплекса со стороны расчёта пространственной ферменной конструкции, её особенности и аспекты. А также была описана работа графической составляющей программного комплекса со скриншотом примера.

Был изучен язык программирования C# для составления программы для автоматизации расчёта и заполнения входного файла для программного комплекса. Была разработана и описана программа для автоматизации расчёта входного файла. В завершении была смоделирована ферменная конструкция на основе заданной цилиндрической оболочки, а также проведены сравнения коэффициента Пуассона реального материала и ферменной конструкции.

Литература

1. Кирсанов М.Н. Аналитический расчет многорешетчатой фермы // Строительная механика и расчет сооружений. 2014. №6. С. 2-6.
2. Александров А.В., Лащеников Б.Я., Шапошников Н.Н. Строительная механика. Тонкостенные пространственные системы Под ред. А.Ф.Смирнова. М.:Стройиздат, 1983. -488 с.
3. Эндрю Стиллмен, Дженнифер Гринм. Изучаем C#. 2012. -704
4. В. В. Субботницкий. Расчёт оптимальных статически неопределимых ферм. 1984. -107
5. В. В. Субботницкий, Б.Н. Иванов. Программный комплекс расчёта статически неопределимых пространственных ферм. 1995. -17
6. А. М. Лукин, Д. А. Лукин, В.В. Квалдыков. Учебно-методическое пособие для студентов заочной и дистанционной форм обучения при подготовке дипломированного специалиста по направлению «СТРОИТЕЛЬСТВО». -86

Приложение

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace ForFERMA
{
    public partial class Form1 : Form
    {
        string txt_write;
        string txt_write_points;
        string txt_write_rods;
        string txt_write_forces;
        delegate void StringArgReturningVoidDelegate(string text);
        volatile bool stop = false;
        int num_rods = 0;
        bool run_again = false;
        public Form1()
        {
            InitializeComponent();
        }

        public void button1_Click(object sender, EventArgs e)
```

```

{
    num_rods = 0;
    txt_write_points = "";
    txt_write_rods = "";
    stop = false;
    buttonClear_Click(sender, e);
    double d;
    txt_write = "";
    if (!double.TryParse(R1.Text, out d))
    {
        MessageBox.Show("Введите корректный внешний радиус цилиндра");
    }
    else
    if (!(double.TryParse(R2.Text, out d)))
    {
        MessageBox.Show("Введите корректный внутренний радиус цилиндра");
    }
    else
    if (!double.TryParse(h.Text, out d))
    {
        MessageBox.Show("Введите корректную длину цилиндра");
    }
    else
    if (!double.TryParse(n.Text, out d))
    {
        MessageBox.Show("Введите корректное количество узлов по окружности цилиндра");
    }
    else
    if (!double.TryParse(n_h.Text, out d))

```

```

{
    MessageBox.Show("Введите корректное количество узлов по длине цилиндра");
}
else
if (!double.TryParse(n_s.Text, out d))
{
    MessageBox.Show("Введите корректное количество узлов по стенке цилиндра");
}
else
if ((Convert.ToDouble(R1.Text) - Convert.ToDouble(R2.Text)) <= 0)
{
    MessageBox.Show("Введите корректные значения радиусов, возможно внутренний
радиус больше либо равен внешнему");
}
else
if (!double.TryParse(force.Text, out d))
{
    MessageBox.Show("Введите корректное значение внешней нагрузки");
}
else
if (!run_again)
{
    label14.Visible = false;
    label15.Visible = false;
    yung.Visible = false;
    label12.Visible = false;
    label13.Visible = false;
    force.Visible = false;
    pictureBox2.Visible = false;
    pictureBox1.Visible = false;
}

```

```
Memo.Visible = true;
progressBar.Visible = true;
buttonClear.Visible = true;
buttonStop.Visible = true;
buttonBack.Visible = true;
label2.Visible = false;
label5.Visible = false;
R1.Visible = false;
h.Visible = false;
pict_cyl.Visible = false;
n.Visible = false;
label6.Visible = false;
buttonGOtxt.Visible = false;
label9.Visible = false;
label10.Visible = false;
label4.Visible = false;
label8.Visible = false;
label11.Visible = false;
n_h.Visible = false;
label1.Visible = false;
n_s.Visible = false;
label3.Visible = false;
label7.Visible = false;
R2.Visible = false;
```

```
Memo.Items.Add("*****");
```

```
Мемо.Items.Add("Вычисление координат узлов для полого цилиндра");
```

```
Мемо.Items.Add("радиусом R = " + R1.Text);
```

```
Мемо.Items.Add("высотой h = " + h.Text);
```



```
Memo.Items.Add("\n*****  
*****");
```

```
    progressBar.Value = 0;
```

```
    progressBar.Maximum = 2*(Convert.ToInt32(n.Text) * Convert.ToInt32(n_h.Text) *  
Convert.ToInt32(n_s.Text));
```

```
    progressBar.Visible = true;
```

```
    double x;
```

```
    double y;
```

```
    double z;
```

```
    double num_n = Convert.ToDouble(n.Text);
```

```
    double num_n_h = Convert.ToDouble(n_h.Text);
```

```
    double num_n_s = Convert.ToDouble(n_s.Text);
```

```
    int num_point = 0;
```

```
    Task.Run(() =>
```

```
    {
```

```
        stop = false;
```

```
        txt_write_rods = "\r\n";
```

```
        txt_write_forces = "\r\n";
```

```
        string zakrep;
```

```
        string dannie = " 0 " + yung.Text + " 1e-4 0 0 0 400e6 0.99 0.99";
```

```
        for (int s = 0; s < num_n_s; ++s)
```

```
        {
```

```
            for (int j = 0; j < num_n_h; ++j)
```

```
            {
```

```
                for (int i = 1; i <= num_n; ++i)
```

```
                {
```

```
                    --i;
```

```
                    if (j != 0)
```

```
                        num_rods += 1;
```

```

if ((s + 1) < (num_n_s))
{
    if (j != 0)
        num_rods += 3;
}

if ((j + 1) < (num_n_h))
{
    num_rods += 2;
}

if ((j + 1) < (num_n_h))
    if ((s + 1) < (num_n_s))
    {
        num_rods += 2;
    }
double l = i;
if (j == 0)
    zakrep = " 1 ";
else
    zakrep = " 0 ";

if ((j != 0) && (s == 0))
    txt_write_forces += (num_point + 1) + " " + Convert.ToDouble(force.Text) *
(-1) * Math.Round((float)Math.Cos(180 * l / (90 * num_n) * Math.PI), 4) + " " +
Convert.ToDouble(force.Text) * (-1) * Math.Round((float)Math.Sin(180 * l / (90 * num_n) *
Math.PI), 4) + " " + j + "\r\n";

x = Math.Cos(180 * (l) / (90 * num_n) * Math.PI) *
(Convert.ToDouble(R1.Text) - (float)(s * (Convert.ToDouble(R1.Text) -
Convert.ToDouble(R2.Text)) / (num_n_s - 1))) + Convert.ToDouble(R1.Text);

```

```

        y = Math.Sin(180 * (l) / (90 * num_n) * Math.PI) *
(Convert.ToDouble(R1.Text) - (float)s * (Convert.ToDouble(R1.Text) -
Convert.ToDouble(R2.Text)) / (num_n_s - 1))) + Convert.ToDouble(R1.Text);
        z = j * (Convert.ToDouble(h.Text) / (num_n_h - 1));
        run_again = true;

        txt_write_points += Convert.ToString(num_point += 1) + zakrep +
Convert.ToString(x) + " " + Convert.ToString(y) + " " + Convert.ToString(z) + "\r\n";

```

```

progressBar.Invoke(
    (ThreadStart)delegate ()
    {
        progressBar.Value += 1;
    });

```

```

Memo.Invoke(
    (ThreadStart)delegate ()
    {
        Memo.Items.Add("x" + (i + 1) + " = " + Convert.ToString(x));
        Memo.Items.Add("y" + (i + 1) + " = " + Convert.ToString(y));
        Memo.Items.Add("z" + (i + 1) + " = " + Convert.ToString(z));
        Memo.SelectedIndex = Memo.Items.Count - 1;
    });

```

```

i++;

```

```

if (i != num_n)

```

```

{

```

```

    if (j != 0)

```

```

        txt_write_rods += Convert.ToString(i + num_n * j + num_n * num_n_h * s)
+ " " + Convert.ToString((i + 1) + num_n * j + num_n * num_n_h * s) + dannie + "\r\n"; // i,j,s
i+1,j,s

```

```

        if ((s + 1) < (num_n_s))
        {
            if (j != 0)
            {
                txt_write_rods += Convert.ToString(i + num_n * j + num_n * num_n_h
* s) + " " + Convert.ToString(i + num_n * j + num_n * num_n_h * (s + 1)) + dannie + "\r\n"; // i,j,s
i,j,s+1

                txt_write_rods += Convert.ToString(i + num_n * j + num_n * num_n_h
* s) + " " + Convert.ToString((i + 1) + num_n * j + num_n * num_n_h * (s + 1)) + dannie + "\r\n"; //
i,j,s i+1,j,s+1

                txt_write_rods += Convert.ToString(i + num_n * j + num_n * num_n_h
* (s + 1)) + " " + Convert.ToString((i + 1) + num_n * j + num_n * num_n_h * s) + dannie + "\r\n"; //
i,j,s+1 i+1,j,s

            }
        }

        if ((j + 1) < (num_n_h))
        {
            txt_write_rods += Convert.ToString(i + num_n * j + num_n * num_n_h *
s) + " " + Convert.ToString(i + num_n * (j + 1) + num_n * num_n_h * s) + dannie + "\r\n"; // i,j,s
i,j+1, s

            txt_write_rods += Convert.ToString(i + num_n * j + num_n * num_n_h *
s) + " " + Convert.ToString((i + 1) + num_n * (j + 1) + num_n * num_n_h * s) + dannie + "\r\n"; //
i,j,s i+1,j+1,s

        }

        if ((j + 1) < (num_n_h))
            if ((s + 1) < (num_n_s))
            {
                txt_write_rods += Convert.ToString(i + num_n * j + num_n * num_n_h
* s) + " " + Convert.ToString(i + num_n * (j + 1) + num_n * num_n_h * (s + 1)) + dannie + "\r\n"; //
i,j,s i,j+1,s+1

```

```

        //txt_write_rods += Convert.ToString(i + num_n * j + num_n *
num_n_h * s) + " " + Convert.ToString((i + 1) + num_n * (j + 1) + num_n * num_n_h * (s + 1)) +
dannie + "\r\n"; // i,j,s i+1,j+1,s+1

        //txt_write_rods += Convert.ToString(i + num_n * j + num_n *
num_n_h * (s + 1)) + " " + Convert.ToString((i + 1) + num_n * (j + 1) + num_n * num_n_h * s) +
dannie + "\r\n"; // i,j,s+1 i+1,j+1,s

        //txt_write_rods += Convert.ToString(i + num_n * (j + 1) + num_n *
num_n_h * s) + " " + Convert.ToString((i + 1) + num_n * j + num_n * num_n_h * (s + 1)) + dannie
+ "\r\n"; // i,j+1,s i+1,j,s+1

        txt_write_rods += Convert.ToString(i + num_n * (j + 1) + num_n *
num_n_h * s) + " " + Convert.ToString((i + 1) + num_n * j + num_n * num_n_h * s) + dannie +
"\r\n"; // i,j+1,s i+1,j,s

        //txt_write_rods += Convert.ToString(i + num_n * (j + 1) + num_n *
num_n_h * (s + 1)) + " " + Convert.ToString((i + 1) + num_n * j + num_n * num_n_h * s) + dannie
+ "\r\n"; // i,j+1,s+1 i+1,j,s
    }
}
else
{
    if (j != 0)

        txt_write_rods += Convert.ToString(i + num_n * j + num_n * num_n_h *
s) + " " + Convert.ToString((i + 1 - num_n) + num_n * j + num_n * num_n_h * s) + dannie + "\r\n";
// i,j,s i+1,j,s

        if ((s + 1) < (num_n_s))
        {
            if (j != 0)
            {
                txt_write_rods += Convert.ToString(i + num_n * j + num_n * num_n_h
* s) + " " + Convert.ToString(i + num_n * j + num_n * num_n_h * (s + 1)) + dannie + "\r\n"; // i,j,s
i,j,s+1

                txt_write_rods += Convert.ToString(i + num_n * j + num_n * num_n_h
* s) + " " + Convert.ToString((i + 1 - num_n) + num_n * j + num_n * num_n_h * (s + 1)) + dannie
+ "\r\n"; // i,j,s i+1,j,s+1
            }
        }
    }
}

```

```

        txt_write_rods += Convert.ToString(i + num_n * j + num_n * num_n_h
* (s + 1)) + " " + Convert.ToString((i + 1 - num_n) + num_n * j + num_n * num_n_h * s) + dannie
+ "\r\n"; // i,j,s+1 i+1,j,s
    }
}

if ((j + 1) < (num_n_h))
{
    txt_write_rods += Convert.ToString(i + num_n * j + num_n * num_n_h *
s) + " " + Convert.ToString(i + num_n * (j + 1) + num_n * num_n_h * s) + dannie + "\r\n"; // i,j,s
i,j+1, s

    txt_write_rods += Convert.ToString(i + num_n * j + num_n * num_n_h *
s) + " " + Convert.ToString((i + 1 - num_n) + num_n * (j + 1) + num_n * num_n_h * s) + dannie +
"\r\n"; // i,j,s i+1,j+1,s
}

if ((j + 1) < (num_n_h))
    if ((s + 1) < (num_n_s))
    {
        txt_write_rods += Convert.ToString(i + num_n * j + num_n * num_n_h
* s) + " " + Convert.ToString(i + num_n * (j + 1) + num_n * num_n_h * (s + 1)) + dannie + "\r\n"; //
i,j,s i,j+1,s+1

        //txt_write_rods += Convert.ToString(i + num_n * j + num_n *
num_n_h * s) + " " + Convert.ToString((i + 1 - num_n) + num_n * (j + 1) + num_n * num_n_h * (s
+ 1)) + dannie + "\r\n"; // i,j,s i+1,j+1,s+1

        //txt_write_rods += Convert.ToString(i + num_n * j + num_n *
num_n_h * (s + 1)) + " " + Convert.ToString((i + 1 - num_n) + num_n * (j + 1) + num_n *
num_n_h * s) + dannie + "\r\n"; // i,j,s+1 i+1,j+1,s

        //txt_write_rods += Convert.ToString(i + num_n * (j + 1) + num_n *
num_n_h * s) + " " + Convert.ToString((i + 1 - num_n) + num_n * j + num_n * num_n_h * (s + 1))
+ dannie + "\r\n"; // i,j+1,s i+1,j,s+1

        txt_write_rods += Convert.ToString(i + num_n * (j + 1) + num_n *
num_n_h * s) + " " + Convert.ToString((i + 1 - num_n) + num_n * j + num_n * num_n_h * s) +
dannie + "\r\n"; // i,j+1,s i+1,j,s

```

```

        //txt_write_rods += Convert.ToString(i + num_n * (j + 1) + num_n *
num_n_h * (s + 1)) + " " + Convert.ToString((i + 1 - num_n) + num_n * j + num_n * num_n_h * s)
+ dannie + "\r\n"; // i,j+1,s+1 i+1,j,s
    }
}
progressBar.Invoke(
    (ThreadStart)delegate ()
    {
        progressBar.Value += 1;
    });
if (stop)
    break;
}
if (stop)
    break;
}
if (stop)
    break;
}

txt_write = Convert.ToString(num_n * num_n_h * num_n_s) + " " +
Convert.ToString(num_rods) + " " + Convert.ToString(num_n * (num_n_h - 1)) + "\r\n";
//Управляющие данные: "число узлов" "число стержней" "число сил"

txt_write += "\r\n";
run_again = false;

buttonGOtxt.Invoke(
    (ThreadStart)delegate ()
    {
        buttonGOtxt.Visible = true;
    });

```

```

buttonGOtxt.Invoke(
    (ThreadStart)delegate ()
    {
        label4.Visible = true;
    });

progressBar.Invoke(
    (ThreadStart)delegate ()
    {
        progressBar.Visible = false;
    });

label4.Invoke(
    (ThreadStart)delegate ()
    {
        label4.Visible = true;
    });

buttonTXT.Invoke(
    (ThreadStart)delegate ()
    {
        buttonTXT.Visible = true;
    });
});
}
}

private void buttonClear_Click(object sender, EventArgs e)
{
    num_rods = 0;
}

```



```

txt_write = "";
txt_write_points = "";
txt_write_rods = "";
Memo.Items.Clear();
}

private void buttonStop_Click(object sender, EventArgs e)
{
    stop = true;
}

private void buttonBack_Click(object sender, EventArgs e)
{
    label14.Visible = true;
    label15.Visible = true;
    yung.Visible = true;
    label12.Visible = true;
    label13.Visible = true;
    force.Visible = true;
    pictureBox2.Visible = true;
    pictureBox1.Visible = true;
    stop = true;
    buttonTXT.Visible = false;
    Memo.Visible = false;
    progressBar.Visible = false;
    buttonClear.Visible = false;
    buttonStop.Visible = false;
    buttonBack.Visible = false;
    label2.Visible = true;
}

```

```

label5.Visible = true;
R1.Visible = true;
h.Visible = true;
pict_cyl.Visible = true;
n.Visible = true;
label6.Visible = true;
buttonGOtxt.Visible = true;
label9.Visible = true;
label10.Visible = true;
label4.Visible = true;
label8.Visible = true;
label11.Visible = true;
n_h.Visible = true;
label1.Visible = true;
n_s.Visible = true;
label3.Visible = true;
label7.Visible = true;
R2.Visible = true;
}

private void buttonTXT_Click(object sender, EventArgs e)
{
    string path = "path";
    string txt_format = ".D";
    bool flag = false;
    int i = 0;
    while (flag == false)
    {
        if (!System.IO.File.Exists(path + i + txt_format))

```

```
    {
        System.IO.File.AppendAllText(path + i + txt_format, txt_write +
txt_write_points.Replace(",", ".") + txt_write_rods + txt_write_forces.Replace(",", "."));
        flag = true;
    }
    else
    {
        i++;
    }
}
}
}
}
```



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»

Инженерная школа

Кафедра механики и математического моделирования

ОТЗЫВ РУКОВОДИТЕЛЯ

на выпускную квалификационную работу студента Мокроусова Игоря Михайловича

направление подготовки 15.03.03 «Прикладная механика» группа Б3409

Руководитель ВКР старший преподаватель кафедры МиММ Ратников А.А.

На тему Моделирование сплошной среды ферменными конструкциями на примере цилиндрической оболочки при действии всестороннего внешнего давления

Дата защиты ВКР «04» июля 2018 г.

Работа посвящена моделированию сплошной среды заменой её на стержневую конструкцию, отвечающей реальному материалу. В ней впервые проведено такое моделирование применительно к криволинейным координатам.

В работе рассматривается осесимметричная задача деформирования цилиндрической оболочки, под действием равномерного внешнего давления. Поскольку решение прямой задачи теории упругости требует интегрирования довольно сложной системы дифференциальных уравнений в частных производных и сопряжено со значительными математическими трудностями, то реальная оболочка заменялась на ферменную конструкцию, а распределённое давление – сосредоточенными силами в узлах.

На языке C# составлен программный код, генерирующий файл с исходными данными (координатами узлов, соединяющими стержнями, приложенной нагрузкой) для программного комплекса расчёта пространственных ферм.

Проведено сравнение решения с полученным ранее методом физической дискретизации. Результаты оказались схожими, что свидетельствует о возможности дальнейшего использования рассмотренного в работе способа расчёта криволинейных тел.

В целом считаю, что студент Мокроусов И.М. достоин присуждения степени бакалавра по направлению подготовки «Прикладная механика», заслуживает оценки «отлично» и может быть рекомендован к поступлению в магистратуру. Заимствование составляет 8 %.

Руководитель ВКР



А.А. Ратников
Уч. степень и звание

А.А. Ратников
подпись

Ратников А.А.
И.О. Фамилия

Собственнору...
«02» июля 2018 г.
тов Ратникова А.А.
уд. ст. тов. ятсн
Список литературы

Маслова Н. Г.