

КАЗАНСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Набережночелнинский институт

А.П. Павленко, В.Н. Никишин

**АНАЛИТИЧЕСКИЕ И ЧИСЛЕННЫЕ МЕТОДЫ
ПРОЧНОСТНОГО АНАЛИЗА И ПРОЕКТИРОВАНИЯ
АВТОМОБИЛЬНЫХ КОНСТРУКЦИЙ**

Допущено

*Учебно-методическим объединением вузов
по университетскому политехническому образованию
в качестве учебного пособия
для студентов высших учебных заведений,
обучающихся по специальности 190109
«Наземные транспортно-технологические средства»*

**КАЗАНЬ
2015**

УДК 519.6 : 629.33.01(075.8)

ББК 39.33.- 02 я 73

П12

Рецензент

кандидат технических наук **Мухаметдинов Э.М.**

Павленко А.П.

П12 Аналитические и численные методы прочностного анализа и проектирования автомобильных конструкций: учеб. пособие для студентов вузов / А.П. Павленко, В.Н. Никишин. – Казань: Изд-во Казан. ун-та, 2015. – 129 с.

ISBN 978-5-00019-375-4

В пособии изложены аналитические и численные методы решения инженерных задач, возникающих в автомобилестроении. Рассмотрены классификация и описания инженерных задач, методов их решения и программного обеспечения. Приведены расчетные выражения методов и примеры решения инженерных задач. Особое внимание уделено методу конечных элементов как основному промышленному методу прочностного анализа автомобильных конструкций.

Издание предназначено для студентов, обучающихся по специальности 190109 «Наземные транспортно-технологические средства», а также магистрантов, аспирантов вузов и специалистов.

УДК 519.6 : 629.33.01(075.8)

ББК 39.33.- 02 я 73

ISBN 978-5-00019-375-4

© А.П.Павленко, В.Н.Никишин, 2015

© Издательство Казанского университета, 2015

СОДЕРЖАНИЕ

Введение	5
Глава I. Общие сведения	8
1.1. Задачи инженерных расчетов.....	8
1.2. Численные методы.....	10
1.3. Прикладное программное обеспечение.....	15
1.4. Методы решения систем линейных алгебраических уравнений, применяемые в инженерных расчетах.....	18
1.5. Аппроксимация полиномами. Численное дифференцирование и интегрирование.....	21
1.6. Определение деформации конструкции применением численных методов для получения аналитических решений дифференциальных уравнений.....	26
1.7. Основные понятия и определения векторно-матричного исчисления.....	31
1.8. <i>Вопросы для самопроверки</i>	35
Глава II. Прочностной анализ автомобильных конструкций	37
2.1. Численные методы анализа.....	37
2.2. Основные понятия математического и функционального анализа. Операторные линейные уравнения и математическая теория их обобщенных решений.....	39
2.3. Введение в вариационное исчисление и метод Ритца.....	43
2.4. Основные положения метода конечных элементов.....	45
2.5. Механическая аналогия между деформацией стержней и интерполяцией сплайнами.....	48
2.6. Задача параметризации и методы её решения.....	51
2.7. <i>Вопросы для самопроверки</i>	55
Глава III. Методы оптимального проектирования автомобильных конструкций	57
3.1. Численные методы оптимизации и минимизации для автомобильных конструкций.....	57
3.2. Метод неопределенных множителей Лагранжа.....	60
3.3. Градиентный метод. Поправки на программном уровне.....	63
3.4. <i>Вопросы для самопроверки</i>	66

Глава IV. Примеры решения задач	68
4.1. Решение задачи анализа на примере растяжения-сжатия стержня.....	68
4.2. Решение задачи оптимального проектирования на примере получения стержня максимальной жесткости на растяжение-сжатие.....	75
4.3. Пример моделирования обвода кузова легкового автомобиля «ВАЗ – 2106» кубическим сплайном кусочно-постоянной жесткости.....	78
4.4. Пример анализа балки передней оси грузового автомо- биля «КАМАЗ – 65115» методом конечных элементов на изгиб.....	97
4.5. Пример определения функции перемещений сечений стержня с линейным законом изменения жесткости на растяжение-сжатие.....	107
4.6. Пример определения точного решения для консольной балки с линейным законом изменения изгибной жесткости....	109
4.7. Пример проектирования поперечной балки основания грузовой платформы автомобиля на основе точного решения уравнения деформации балки.....	112
4.8. <i>Вопросы для самопроверки</i>	125
Список рекомендуемой литературы	128

ВВЕДЕНИЕ

Создание новой автомобильной техники является длительным дорогостоящим наукоемким и трудоемким процессом. В условиях жесткой конкуренции стремятся сократить сроки разработки и затраты, что невозможно без применения современных высокоэффективных средств автоматизации труда инженеров. С этой целью сегодня широко используются системы автоматизированного проектирования (САПР), область применения которых давно вышла за рамки проектирования и охватывает практически весь жизненный цикл изделия (автомобиля): от маркетинговых исследований до утилизации.

Рост производительности труда при одновременном увеличении качества проектно-конструкторских работ обеспечивается за счет компьютеризации. Применение ЭВМ в качестве орудия труда обуславливает появление новых форм мыслительной и творческой деятельности, что можно рассматривать как историческое развитие человечества, и предполагает переориентацию на данное средство. В связи с этим развитие вычислительной техники объясняет интенсификацию работ по созданию новых методов во всех отраслях науки, ориентированных на ЭВМ.

Современные САПР предполагают применение ЭВМ и развиваются в направлении комплексной автоматизации на всех этапах жизненного цикла изделия, что также актуализирует проблему создания новых методов проектирования и конструирования, математического и программно-алгоритмического обеспечения для решения различных инженерных задач. Разработка и применение таких методов значительно снижает затраты на проектирование и технологическую подготовку производства объектов автомобилестроения за счет снижения сроков их разработки.

Вместе с тем новые методы накладывают свой отпечаток. Такие методы должны быть формализованными, то есть их алгоритмы должны без особых трудностей записываться на языках программирования для ЭВМ. Неформализуемые методы часто отбрасываются как неэффективные при использовании САПР, и наоборот, удачные и эффективные алгоритмы без больших затрат переносятся в САПР других отраслей или при разработке

САПР для более современных компьютерных аппаратных платформ. При этом возможен эффективный автоматический или автоматизированный перевод алгоритмов на другие языки программирования, так как алгоритмические языки имеют преемственность, даже если имеет место их изменение или совершенствование.

Проблема формализации решалась применением математического аппарата. Однако не всякий математический метод может быть переведен на язык компьютера. Фактически ЭВМ может самостоятельно выполнять лишь некоторые арифметические действия над натуральными числами (складывать и вычитать) и логические операции сравнения (больше, меньше). Разумеется, другие операции компьютер также выполняет, но, по большому счету, благодаря программному обеспечению, которое сводит высшие операции к простейшим ранее упомянутым. Например, операция умножения сводится к суммированию одного из множителей столько раз, пока это количество раз не превысит значения второго множителя. При этом немалую роль играет математическое обеспечение, на основе которого создавалось данное программное обеспечение. Собственно говоря, такие приемы математики выделили в отдельную отрасль – численные методы, задача которых все же не является формализовать математический метод, а лишь получить частное решение конкретной математической задачи. Задачу формализации следует уже отнести к программированию, которое на данный момент по-прежнему, по мнению многих исследователей, остается искусством.

Несомненно, для решения важных инженерных задач, требующих значительного объема вычислений и конкретного результата за отведенное для этого время, применяются численные методы. При этом расчетчик не всегда обладает достаточной информацией о решаемой задаче, и не всегда удается получить результат в срок или вообще его получить. Тогда решение все же принимается на основании неполных данных или волею руководителя. Тем не менее, накоплен некоторый опыт при решении инженерных задач, который и составляет предмет дисциплины «Аналитические и численные методы решения уравнений математической физики».

В данном учебном пособии не ставилась цель перечислить и изложить все математические методы, применяющиеся в инженерных расчетах. Понятно, что это сделать не в силах в рамках данной работы, да и нет в этом большого смысла. При необходимости можно воспользоваться специальной математической литературой по численным методам. Пособие ставит целью научить применять математические методы и вычислительную технику для решения инженерных задач. Практические задачи не всегда могут быть решены типовым набором численных методов и, тем более, функциями и алгоритмами отдельных САПР, поэтому их приходится решать всеми имеющимися средствами, в том числе и вручную. Накопленный опыт позволяет выделить важное на сегодняшний день в данной дисциплине, что способствует подготовке в автомобилестроении современных инженеров, способных быстро и надежно решать поставленные перед ними задачи с применением высокоэффективных средств автоматизации вычислений.

Изложенные в данном пособии методы призваны оказывать максимальную помощь при решении практических задач, действовать развитию творческой мысли, но никак не быть шаблоном. Вопреки предвзятому мнению, в численных методах и сейчас остается много «белых пятен», есть еще много неясного, нерешенного, много задач еще ждут своей формализации. И нельзя все это переложить на плечи математиков, прикладные вопросы в численных методах развиваются из потребностей практики. Только инженер способен корректно сформулировать стоящую перед ним инженерную задачу и критерии оценки правильности ее решения. При этом он располагает множеством физических аналогий, чего не имеет математик. Как правило, легче получить решение, правильность которого доказывается потребностью, например, получением в качестве решения конструкции, которая лучше ранее применяемой, чем математически это доказать. Дисциплина «Аналитические и численные методы решения уравнений математической физики» дает высокоэффективный современный инструмент, позволяющий решать практические задачи, стоящие перед инженером, а также открывает возможности его адаптации к конкретным задачам и совершенствованию.

ГЛАВА I. ОБЩИЕ СВЕДЕНИЯ

1.1. Задачи инженерных расчетов

Большинство инженерных задач, представляющих интерес отдельного изучения, сводятся к решению дифференциальных уравнений. Дифференциальными уравнениями описываются силовые конструкции.

Силовой конструкцией является та часть конструкции, которая непосредственно участвует в восприятии нагрузки. Геометрические размеры силовых конструкций выбираются исходя из обеспечения критериев прочности, жесткости, устойчивости, долговечности и других подобных критериев. Примером несилевой конструкции может служить какая-нибудь ручка, геометрические размеры которой выбираются исходя из удобства ее обхвата человеческой рукой, а не из вышеназванных критериев.

Конструкция описывается рядом параметров, входящих в дифференциальное уравнение этой конструкции, а также различными ограничениями на данные параметры, называемыми краевыми условиями. К числу этих параметров относятся:

- проектные параметры;
- параметры перемещений (или деформаций);
- параметры нагрузок (или напряжений).

В качестве **проектных параметров** берутся геометрические размеры конструкции и механические характеристики материалов, а также их производные характеристики: площади поперечных сечений, массовые или жесткостные характеристики, в комплексе учитывающие свойства материала и геометрические размеры.

Параметры нагрузок включают внешние нагрузки в виде распределенной нагрузки или сосредоточенных сил и моментов, внутренние силовые факторы (осевые и поперечные силы, изгибающие и крутящие моменты), реакции, напряжения (нормальные и касательные).

Параметры перемещений состоят из обобщенных перемещений (линейных и угловых) или деформаций.

Параметры перемещений и параметры нагрузок могут зависеть от проектных параметров, а также друг от друга. Поэто-

му существует множество *форм инженерных задач*, которые можно разделить на:

- задачи в форме метода сил;
- задачи в форме метода перемещений.

Следует отметить, что *метод сил* требует высокой квалификации расчетчика и при неудачном выборе расчетной схемы может привести к значительным погрешностям в численном решении, поэтому наиболее универсальным и широко применяемым является метод перемещений.

По *методу перемещений* в качестве параметров нагрузки берется внешняя нагрузка, независящая от проектных параметров и обычно составляющая правую (свободную) часть дифференциального уравнения деформации конструкции. Само дифференциальное уравнение отражает связь между заданной внешней нагрузкой и перемещениями в зависимости от проектных параметров. Проектные параметры и параметры перемещений входят в левую часть уравнения.

В дифференциальное уравнение деформации конструкции входят известные и неизвестные параметры, причем неизвестные параметры подлежат определению через известные параметры. Рассмотренные виды параметров, за исключением заданных параметров нагрузки, могут быть как известными, так и неизвестными. В зависимости от того, какие параметры заданы или известны, а какие подлежат определению, инженерные задачи разделяются на следующие *постановки*:

- задачи инженерного анализа;
- задачи оптимального проектирования (оптимизации).

Под *задачей инженерного анализа* понимается определение компонентов напряженно-деформированного состояния (НДС) конструкции. Задачу анализа в форме метода перемещений решают при заданных проектных параметрах и внешней нагрузке, определяя при этом неизвестные параметры перемещений – компоненты деформированного состояния. Зная перемещения, далее методами теории упругости несложно определяются деформации и напряжения.

Задача оптимального проектирования состоит в отыскании проектных параметров, удовлетворяющих определенным условиям, зависящим от цели оптимизации конструкции. При

этом параметры перемещений также остаются неизвестными, поэтому решение задачи проектирования ведется последовательными приближениями – итерационным процессом, в ходе которого приходится неоднократно решать задачу анализа конструкции.

При решении различных инженерных задач возникает *проблема задания геометрии конструкции*. Для этого применяются методы, полученные на основе аналогии с задачами инженерного анализа. Например, широко применяемый в САПР для задания сложной геометрии конструкции кубический сплайн аналогичен гибкой плазовой рейке постоянной изгибной жесткости, нагруженной внутренними реакциями, вызванными заданными смещениями опор, и не испытывающей внешней нагрузки. Следовательно, кубический сплайн является решением однородного дифференциального уравнения деформации балки (правая часть такого уравнения равна нулю ввиду отсутствия на сплайн внешней нагрузки), при этом частично заданы параметры перемещений, а остальные перемещения подлежат определению. Поэтому задачи задания геометрии конструкции сплайнами и задачи инженерного анализа решаются по единому численному алгоритму.

1.2. Численные методы

Численные методы – это прикладной раздел математики, изучающий вопросы разработки и применения методов для получения решения математических задач за конечное число вычислений. Такие решения должны без особых трудностей доводиться до численного результата. Следует отметить, что решение задачи может быть как в аналитическом виде, то есть в виде математических выражений, так и в численном. Конечное число вычислений гарантирует принципиальную возможность довести аналитическое решение до численного результата. Так же можно отметить, что аналитические решения существуют, но не всегда их удается выразить через известные математические функции, или они могут иметь неприемлемо громоздкий вид, поэтому на практике часто обходят задачу получения аналитического

решения, довольствуясь только численными значениями результата.

Численные методы также известны как прикладная или вычислительная математика, численный анализ, научное вычисление, компьютер в математических исследованиях и тому подобное. Их развитие часто связывают с развитием ЭВМ, что не совсем правильно. Численные методы появились задолго до появления первых компьютеров и первоначально применялись для ручного счета. Развитие методов вычислений приводило к совершенствованию техники счета, внедрению в нее новых изобретений, таких как счеты (абак), арифмометр, логарифмическая линейка, калькулятор и, наконец, компьютер. Разумеется, появление нового средства вычислений переориентировало на себя и методы вычислений. Например, появление калькуляторов увело в историю численные методы арифметических операций, таких как всем известное выполнение арифметических операций в столбик. С появлением компьютеров дело стало обстоять сложнее. Современные программы для ЭВМ уже способны оперировать не только численными данными, но и символьными, а следовательно, выполнять вычисления в аналитическом виде. Это посягает на отбрасывание более сложных вычислительных приемов, часто имеющих отношение к построению математической теории, например, такого приема как интегрирование по частям, что очевидно при решении простых задач высшей математики на ЭВМ. Однако при решении сложных задач инженерного анализа становится ясной поспешность данного вывода ввиду несовершенства программного обеспечения и вычислительной техники, а также отсутствия изученности некоторых вопросов их применения. ЭВМ не заменяют творческого труда человека при постановке задач и выбора методов решения, а лишь позволяют решать задачи, недоступные для решения другими средствами.

Тем не менее, современным эффективным средством вычислений, ориентирующим на себя численные методы, на сегодня являются ЭВМ. Успех применения компьютеров привел к выделению численных методов из различных отраслей математики в отдельную математическую дисциплину. Современные численные методы ориентированы на применение ЭВМ и по-

зволяют решать задачи с большим числом неизвестных. По структуре раздел математики «Численные методы» состоит из упорядоченного набора различных методов, предназначенных для решения определенных *видов математических задач*, например, методы:

- решения систем линейных алгебраических уравнений;
- решения нелинейных алгебраических уравнений;
- численного дифференцирования и интегрирования;
- аппроксимации и интерполяции;
- оптимизации и минимизации;
- решения дифференциальных уравнений в обыкновенных производных;
- решения дифференциальных уравнений в частных производных.

Разнообразие численных методов объясняется различным решением вопроса нахождения компромисса между противоречивыми требованиями универсальности, точности, надежности и вычислительной стоимости, предъявляемым к методам. При решении конкретной практической задачи необходимо выбрать из всего многообразия методы, наиболее полно подходящие для получения ее решения, удовлетворяющего заданной точности. При этом решение должно быть получено к заданному сроку располагаемыми средствами – вычислительными ресурсами. При решении нескольких однотипных задач различной сложности руководствуются также требованиями универсальности и надежности.

Выбор численных методов и построение из них последовательной логической цепочки для решения конкретной задачи составляет *стратегию численных методов*. В самом деле, предполагается, что непосредственно мы можем решить лишь систему линейных алгебраических уравнений с нижней треугольной матрицей вида:

1.3. Прикладное программное обеспечение

Как уже говорилось, современные численные методы предполагают применение ЭВМ. Это требует от расчетчика навыков программирования. Чтобы воспользоваться методом, необходимо реализовать его алгоритм в виде программы для ЭВМ. Накопленный опыт в этой сфере позволил выделить реализации наиболее удачных и эффективных алгоритмов в отдельное программное обеспечение, значительно облегчающее использование численных методов. Тем не менее, реально стоящая на практике задача не всегда может быть решена типовым набором численных методов, хотя и может их частично включать. Поэтому также необходимо располагать знаниями об имеющемся программном обеспечении и его возможностях, чтобы не концентрировать внимание на отладке хорошо известных алгоритмов типовых численных методов.

Несомненно, наиболее универсальным средством являются *языки программирования высокого уровня*. Для численных расчетов широко распространены языки C и C++ [3, 14]. Это объясняется не только их высокой универсальностью по сравнению с другими языками программирования, а также тем, что на данных языках написаны операционные системы, обеспечивающие взаимодействие с аппаратной частью компьютера, и различные вставки – части текста прикладных программ, которые по тем или иным причинам не могли быть написаны на языке реализации данных программ. Программы, написанные на языках C и C++, обладают высоким быстродействием и совместимостью. К недостаткам языков программирования следует отнести необходимость сведения алгоритмов численных методов непосредственно к арифметическим операциям и трудности их отладки, например, в некоторых языках даже нет оператора возведения в степень. Поэтому требуется высокая квалификация пользователя.

Более подходящим средством для вычислений является специализированное программное обеспечение. Мощнейшим инструментом численных расчетов является *вычислительная среда* MATLAB [1,12], обладающая встроенным специализированным языком программирования, в котором заранее опреде-

лены и выполняются операции над векторами и матрицами, встроенными операторами решаются системы линейных алгебраических уравнений. Здесь имеется множество встроенных функций, реализующих алгоритмы численных методов и визуализирующих числовые данные (построение графиков), а также инструментов отладки алгоритмов. Недостатком является необходимость представления математических выражений в компьютерном виде (в одну строку ограниченным набором символов), что затрудняет проверку правильности их набора. Что касается времени вычислений, то по быстрдействию встроенные функции сравнимы с программами, написанными на языках программирования, а алгоритмы, записанные на встроенном языке MATLAB, значительно им уступают. Часто возникают случаи, когда более эффективный метод, записанный на языке MATLAB, работает дольше, чем менее эффективный, но вызванный встроенной функцией.

Для получения аналитических решений имеются *среды символьной математики* Maple и Mathcad. Более удобным интерфейсом работы с пользователем обладает Mathcad [23], он же представляет выражения в математическом виде, удобном для человеческого восприятия и отладки алгоритмов. Область применения этой системы заключается в получении несложных аналитических выкладок и проведении прикладных расчетов. Для сложных выкладок среды символьной математики непригодны и не могут всецело заменить человека. Творчество расчетчика заключается в том, чтобы разбить сложные выкладки на ряд простых, которые уже возможно получить и вручную, по аналогии с вычислениями, чтобы иметь возможность применить имеющиеся средства. А для этого необходимо знать и уметь применять различные вычислительные приемы, такие как подстановка или интегрирование по частям. Несмотря на то, что те выкладки, которые способны выполнить среды символьной математики, можно получить и вручную, их применение объясняют следующие преимущества: символьные вычисления на ЭВМ проводятся гораздо быстрее и без ошибок, при этом не требуется пользоваться справочной литературой, что также экономит время.

Система Maple [9] более надежно решает аналитические задачи и также может представлять выражения в математическом виде. Однако набор выражений приходится проводить в компьютерном виде, и только затем их можно перевести в математический вид, что не совсем удобно. Многие функции Maple встроены в MATLAB, тем не менее, использовать их в Maple более удобно.

Что касается вычислений в средах символьной математики, то в них также можно получать и численные результаты. Они имеют встроенные функции, могут визуализировать результаты, позволяют программировать. Хотя функции программирования записываются достаточно сложно и неудобно в сравнении с языками программирования. Большое количество типов данных приводит к зависанию и прерыванию алгоритмов. При этом среды символьной математики имеют наименьшее быстродействие. Поэтому для практических вычислений они малопригодны.

Таким образом, для получения расчетных выражений для методов, отладки небольших блоков алгоритмов, проведения прикладных расчетов следует применять среды символьной математики. Для формирования окончательной расчетной программы, получения и представления результатов необходимо применять мощные вычислительные среды. И если необходимо постоянно решать однотипные задачи с большим количеством вычислений, то алгоритм расчетной программы необходимо перенести на язык программирования.

Отметим, что в САПР имеются **системы автоматизации инженерных расчетов** (CAE-системы), обладающие набором встроенных функций численного решения инженерных задач. Особенностью большинства из них является то, что инженер-расчетчик – пользователь данной системы – непосредственно не имеет дела с численными методами. Он лишь подготавливает данные, запрашиваемые программой в диалоговом режиме, и получает конечный результат. Таким образом производители таких систем пытаются снизить требования к квалификации расчетчика. Тем не менее, понимание алгоритмов численных методов способствует получению правильных и достаточно точных решений при минимальных вычислительных затратах и делает более осознанным диалог инженера с системой.

В общем случае $n \neq m$, то есть число уравнений не совпадает с числом неизвестных, однако такие задачи возникают на практике, например, при обработке результатов эксперимента с недостаточным или избыточным количеством измерений. При недостаточном количестве измерений ($n < m$) находят зависимость n неизвестных от $m - n$ независимых (базисных) неизвестных. При избыточном количестве измерений с погрешностями $n > m$ решают задачу выравнивания (осреднения) с целью нахождения неизвестных с минимальными погрешностями.

Система (1.3) имеет не более одного решения при одинаковом числе уравнений и неизвестных ($n = m$). Теоретически решение можно записать как $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, где \mathbf{A}^{-1} – обратная матрица. Явное теоретическое обращение матрицы \mathbf{A} требует астрономического количества вычислений. Для получения решения системы при $n = m$ (1.3) не нужно явно обращать матрицу, достаточно всего лишь преобразовать ее в систему (1.1) с нижней треугольной матрицей, решение для которой нам известно. С этой целью широко применяется *метод Гаусса*, суть которого заключается в следующем: на первом шаге каждое уравнение делят на свой коэффициент при первом неизвестном $a_{i,0}$, с тем чтобы коэффициенты при первом неизвестном во всех уравнениях равнялись единице; затем первое уравнение вычитается из остальных уравнений, тогда все элементы первого столбца матрицы, кроме первого элемента, становятся равными нулю; затем шаги последовательно продолжают для следующих уравнений и коэффициентов до тех пор, пока не дойдут до последнего коэффициента в последнем уравнении; в итоге система (1.3) преобразуется к системе с верхней треугольной матрицей, диагональные коэффициенты которой равны единице – это прямой ход метода Гаусса. Система с верхней треугольной матрицей может быть решена обратной подстановкой, называемой в данном случае обратным ходом метода Гаусса.

Однако при прямом ходе изложенного метода может возникнуть ситуация, когда элемент матрицы (коэффициент), лежащий на диагонали, будет равен нулю. Деление на ноль выдаст ошибку и прерывание алгоритма. Для предотвращения такого случая меняют уравнения в системе и порядок индексов неиз-

вестных местами, с тем чтобы после каждого шага на диагонали матрицы системы оставался наибольший по модулю элемент соответствующего столбца или строки матрицы. Это называется **выбором ведущего (главного) элемента**. При этом, как правило, снижаются погрешности вычислений. Отметим, что менять уравнения местами гораздо проще, чем неизвестные, поэтому для снижения вычислительных затрат чаще применяется метод Гаусса с выбором ведущего (главного) элемента столбца.

Есть еще одно замечание: когда элемент матрицы равен нулю, то деление его на другой коэффициент также даст ноль, поэтому для сокращения вычислений рассматривать такие элементы нет необходимости. Учет в алгоритме метода Гаусса структуры матрицы для сокращения вычислительных затрат с тем чтобы не рассматривать нулевые элементы, приводит к его модификациям (разновидностям). Отметим, что в инженерном анализе особую роль играют системы с ленточной структурой матрицы, с так называемыми n -диагональными матрицами, для которых следует применять метод Гаусса с выбором ведущего элемента столбца для разреженных матриц.

Широко применяются в инженерных задачах и проще всего решаются системы с трехдиагональными матрицами. Метод Гаусса в этом случае вырождается в **метод алгебраической прогонки**. В этом случае не следует хранить в памяти ЭВМ квадратную матрицу, три ненулевые диагонали коэффициентов вносят в три массива A , B , C и четвертый массив D для правых частей уравнений, тогда система уравнений может быть записана в следующем виде:

$$A_i x_{i-1} - B_i x_i + C_i x_{i+1} = D_i, \quad i = 0, 1, \dots, n, \quad (1.4)$$

где A – нижняя диагональ; B – главная диагональ; C – верхняя диагональ; элементы $A_0 = 0$ и $C_n = 0$, так как они выходят за границы квадратной матрицы системы.

Решение отыскивается в виде обратного хода:

$$x_i = \xi_{i+1} x_{i+1} + \eta_{i+1},$$

после его подстановки в (1.4), определяются вспомогательные массивы ξ и η .

Алгоритм прогонки начинается с развязки, где несуществующие элементы приравниваются к нулю:

$$A_0 = 0, C_n = 0, \xi_0 = 0, \eta_0 = 0, x_{n+1} = 0.$$

Необходимость привлечения таких элементов вызвана единообразием записи системы и решения. Затем выполняется прямой ход, в котором вычисляются вспомогательные массивы

$$\xi_{i+1} = C_i / (B_i - A_i \xi_i), \eta_{i+1} = (A_i \eta_i - D_i) / (B_i - A_i \xi_i), \\ i = 0, 1, \dots, n.$$

Далее выполняется обратный ход, в котором вычисляются искомые неизвестные:

$$x_i = \xi_{i+1} x_{i+1} + \eta_{i+1}, i = n, n-1, \dots, 0.$$

Методы решения систем линейных алгебраических уравнений не исчерпываются рассмотренными методами. Большинство из них направлено на преобразование матрицы системы с целью выполнения каких-либо операций над матрицами или получения матриц заданной структуры. Такие методы объединены в численные методы векторной алгебры (линейной алгебры и векторной геометрии) и могут быть применены для решения линейных систем уравнений. Практически все они реализованы в прикладном программном обеспечении.

1.5. Аппроксимация полиномами.

Численное дифференцирование и интегрирование

Большинство зависимостей, используемых в практических расчетах, не может быть описано точными функциями, так как класс известных функций весьма ограничен. Иногда точные функции настолько сложны, что возникает вопрос их упрощения. Здесь возникает задача *аппроксимации (приближения)* – задание кривой в аналитическом виде, то есть в виде математических выражений. Реальная функция заменяется близкой по значениям, но более простой по выражению аппроксимирующей функцией, то есть приближается или аппроксимируется.

Наиболее простой метод аппроксимации – это замена функции на полином (многочлен), так как полиномы хорошо изучены и просты по форме. Аппроксимация функции $f(x)$ по-

линомом $P_n(x)$ степени n может быть записана следующим образом:

$$f(x) \cong P_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n. \quad (1.5)$$

Здесь коэффициенты полинома a_i , $i = 0, 1, \dots, n$ являются неизвестными, то есть задача аппроксимации полиномами сводится к отысканию неизвестных коэффициентов аппроксимирующего полинома. Сходной с задачей аппроксимации является задача **интерполяции** – проведение гладкой кривой по точкам. И неизвестные коэффициенты аппроксимирующего полинома отыскиваются из условий интерполирования, суть которых заключается в том, чтобы аппроксимирующий полином $P_n(x)$ точно совпадал с аппроксимируемой функцией $f(x)$ в заданных точках при x_i . Условия интерполирования можно записать следующим образом:

$$f(x_i) = P_n(x_i) \cong a_0 + a_1x_i + a_2x_i^2 + \dots + a_nx_i^n, \quad i = 0, 1, \dots, n. \quad (1.6)$$

Условия (1.6) образуют систему линейных алгебраических уравнений относительно искомым коэффициентов полинома a_i , $i = 0, 1, \dots, n$, которую можно записать в следующем алгебраическом и матричном виде:

$$\begin{cases} a_0 + a_1x_0 + a_2x_0^2 + \dots + a_nx_0^n = f(x_0) \\ a_0 + a_1x_1 + a_2x_1^2 + \dots + a_nx_1^n = f(x_1) \\ a_0 + a_1x_2 + a_2x_2^2 + \dots + a_nx_2^n = f(x_2) \\ \dots \\ a_0 + a_1x_n + a_2x_n^2 + \dots + a_nx_n^n = f(x_n) \end{cases} \Rightarrow$$

$$\Rightarrow \begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \dots \\ a_n \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \\ \dots \\ f(x_n) \end{bmatrix}. \quad (1.7)$$

Отметим, что количество заданных точек – количество условий интерполирования – влияет на степень полинома. На практике применяются полиномы степени не выше 5, так как при более высоких степенях графики полиномов осциллируют – нежелательно изгибаются (появляется волнистость).

Матрица системы (1.7) имеет особую структуру и называется матрицей Вандермонда. На практике эту систему не решают, а используют полиномы Лагранжа и Ньютона, в которых неизвестные коэффициенты уже явно выражены через решение этой системы.

Многочлен Лагранжа записывается в следующем виде:

$$P_n(x) = \sum_{i=0}^n \frac{\prod_{j=0, j \neq i}^n (x - x_j)}{\prod_{j=0, j \neq i}^n (x_i - x_j)} f(x_i). \quad (1.8)$$

Многочлен Ньютона записывается в следующем виде:

$$P_n(x) = \sum_{i=0}^n f^{(n)}(x_i) \prod_{j=0}^{i-1} (x - x_j). \quad (1.9)$$

О численном получении производных $f^{(n)}(x_i)$ речь пойдет ниже.

Преимуществом многочлена Ньютона является то, что при добавлении новой точки не нужно пересчитывать весь многочлен, достаточно вычислить дополнительный член (рекуррентное свойство). Так как используются многочлены низких степеней (не выше 5) и вычислительные машины, то это преимущество теряет свою важность, и на практике чаще всего используют многочлен Лагранжа.

Как уже отмечалось, количество заданных точек (значений функции) влияет на степень полинома, но на практике для устранения осцилляций применяются полиномы низких степеней (не выше 5). Поэтому функция может быть аппроксимирована участками, включающими по несколько ее значений. Зная коэффициенты аппроксимирующего полинома (1.5), нетрудно вычислить его производные и интегралы, близкие к производным и интегралам аппроксимируемой функции:

$$\frac{df(x)}{dx} \equiv f'(x) \equiv \frac{dP_n(x)}{dx} \equiv P_n'(x) = a_1 + 2a_2x + \dots + na_nx^{n-1};$$

$$\int f(x)dx \cong \int P_n(x)dx = a_0x + \frac{a_1}{2}x^2 + \dots + \frac{a_n}{n+1}x^{n+1} + C, \quad (1.10)$$

где C – постоянная интегрирования.

Таким образом, задачи **численного дифференцирования и интегрирования** функции сводятся к задаче ее аппроксимации полиномами, отысканию неизвестных коэффициентов полиномов и непосредственному вычислению производных и интегралов по их выражениям.

Рассмотрим некоторые частные примеры. Функцию $f(x)$, заданную на сетке значений ее аргумента x , называемой сеткой $\Delta: a = x_0 < x_1 < \dots < x_n = b$, будем аппроксимировать полиномом Лагранжа первой степени по участкам $[x_{i-1}, x_i]$, $i = 1, 2, \dots, n$:

$$f(x) \cong P_1(x) = \frac{(x - x_i)}{(x_{i-1} - x_i)} f(x_{i-1}) + \frac{(x - x_{i-1})}{(x_i - x_{i-1})} f(x_i),$$

$$\forall x \in [x_{i-1}, x_i], \quad i = 1, 2, \dots, n. \quad (1.11)$$

Полином Лагранжа первой степени описывает ломаную, последовательно соединяющую заданные значения аппроксимируемой функции прямыми отрезками. Формула для определения производной выражения (1.11) называется **конечной разностью** и соответствует определению производной без предела:

$$f'(x) \cong P'_1(x) = \frac{f(x_i) - f(x_{i-1})}{(x_i - x_{i-1})}, \quad \forall x \in [x_{i-1}, x_i], \quad i = 1, 2, \dots, n. \quad (1.12)$$

В численных методах конечными разностями с определенной степенью точности заменяют непрерывные производные, например в формуле для многочлена Ньютона (1.9).

Определенный интеграл выражения (1.11) геометрически может быть истолкован как площадь, ограниченная графиком прямой линии, что образует трапецию, поэтому определенный интеграл выражения (1.11) называют **формулой трапеции** и записывают в виде:

$$\int_{x_{i-1}}^{x_i} f(x)dx \cong \int_{x_{i-1}}^{x_i} P_1(x)dx = \frac{1}{2}[f(x_i) + f(x_{i-1})](x_i - x_{i-1}),$$

$$i = 1, 2, \dots, n. \quad (1.13)$$

Если требуется вычислить определенный интеграл по нескольким участкам, то достаточно просуммировать значения интегралов соответствующих участков:

$$\begin{aligned} \int_{a=x_0}^{b=x_n} f(x)dx &\cong \sum_{i=1}^n \int_{x_{i-1}}^{x_i} P_1(x)dx = \\ &= \frac{1}{2} \sum_{i=1}^n [f(x_i) + f(x_{i-1})](x_i - x_{i-1}). \end{aligned} \quad (1.14)$$

При аппроксимации функции полиномом Лагранжа второй степени:

$$\begin{aligned} f(x) \cong P_2(x) &= \frac{(x-x_i)(x-x_{i+1})}{(x_{i-1}-x_i)(x_{i-1}-x_{i+1})} f(x_{i-1}) + \\ &+ \frac{(x-x_{i-1})(x-x_{i+1})}{(x_i-x_{i-1})(x_i-x_{i+1})} f(x_i) + \frac{(x-x_{i-1})(x-x_i)}{(x_{i+1}-x_{i-1})(x_{i+1}-x_i)} f(x_{i+1}), \\ &\forall x \in [x_{i-1}, x_{i+1}], \quad i = 1, 2, \dots, n-1, \end{aligned} \quad (1.15)$$

выражение определенного интеграла называется **формулой Симпсона**:

$$\begin{aligned} \int_{x_{i-1}}^{x_{i+1}} f(x)dx &\cong \int_{x_{i-1}}^{x_{i+1}} P_2(x)dx = \frac{1}{3} [f(x_{i-1}) + 4f(x_i) + f(x_{i+1})]d, \\ &i = 1, 2, \dots, n-1, \end{aligned} \quad (1.16)$$

причем для упрощения формулы используется равный шаг по аргументу между задаваемыми значениями функции: $d = (x_i - x_{i-1}) = (x_{i+1} - x_i)$.

При аппроксимации функции полиномом Лагранжа третьей степени $P_3(x)$ выражение определенного интеграла называется

формулой Симпсона $\frac{3}{8}$:

$$\begin{aligned} \int_{x_{i-1}}^{x_{i+2}} f(x)dx &\cong \int_{x_{i-1}}^{x_{i+2}} P_3(x)dx = \\ &= \frac{3}{8} [f(x_{i-1}) + 3f(x_i) + 3f(x_{i+1}) + f(x_{i+2})]d, \quad i = 1, 2, \dots, n-2. \end{aligned} \quad (1.17)$$

При аппроксимации функции полиномом Лагранжа четвертой степени $P_4(x)$ выражение определенного интеграла называется **формулой Буля**:

$$\int_{x_{i-1}}^{x_{i+3}} f(x)dx \cong \int_{x_{i-1}}^{x_{i+3}} P_4(x)dx =$$

$$= \frac{2}{45} [7f(x_{i-1}) + 32f(x_i) + 12f(x_{i+1}) + 32f(x_{i+2}) + 7f(x_{i+3})]d,$$

$$i = 1, 2, \dots, n - 3. \quad (1.18)$$

1.6. Определение деформации конструкции применением численных методов для получения аналитических решений дифференциальных уравнений

Как уже отмечалось, целью численных методов является сведение задачи к такой задаче, для которой известно аналитическое решение. Важные для инженерных расчетов задачи решения дифференциальных уравнений деформации сложных конструкций численными методами ранее сводились к дифференциальным уравнениям деформации простых конструкций постоянной жесткости, для которых известны аналитические решения. Современные численные методы позволяют решать в аналитическом виде дифференциальные уравнения деформации конструкций переменной жесткости, с тем чтобы к данному решению можно было свести задачу инженерного анализа. Рассмотрим данный вопрос на примере стержневых конструкций, работающих на растяжение-сжатие, кручение, поперечный изгиб. Напомним, что стержнем является элемент конструкции, у которого продольный размер (длина) значительно больше двух поперечных (высоты и ширины).

Дифференциальное уравнение деформации стержня, работающего на растяжение-сжатие, записывается следующим образом:

$$- [EA(x)u'(x)]' = r(x), \quad \forall x \in (a, b), \quad (1.19)$$

где $EA(x)$ – функция жесткости на растяжение-сжатие;

$u(x)$ – искомая функция линейных перемещений поперечных сечений стержня;

$r(x)$ – распределенная (погонная) внешняя растягивающая нагрузка; ось Ox направлена вдоль оси стержня.

Дифференциальное уравнение деформации стержня, работающего на кручение, аналогично и записывается следующим образом:

$$- [GI_p(x)\gamma'(x)]' = m(x), \quad \forall x \in (a, b), \quad (1.20)$$

где $GI_p(x)$ – функция жесткости на кручение;

$\gamma(x)$ – искомая функция угловых перемещений поперечных сечений стержня;

$m(x)$ – распределенный (погонный) внешний крутящий момент; ось Ox направлена вдоль оси стержня.

Уравнения (1.19) и (1.20) аналогичны и могут быть записаны одним общим уравнением:

$$- [C(x)u'(x)]' = p(x), \quad \forall x \in (a, b), \quad (1.21)$$

где $C(x)$ – функция жесткости;

$u(x)$ – искомая функция перемещений;

$p(x)$ – распределенная (погонная) внешняя нагрузка; ось Ox направлена вдоль оси стержня.

Дифференциальное уравнение деформации стержня, работающего на поперечный изгиб, записывается следующим образом:

$$[EI(x)v''(x)]'' = q(x), \quad \forall x \in (a, b), \quad (1.22)$$

где $EI(x)$ – функция жесткости на изгиб;

$v(x)$ – искомая функция прогибов;

$q(x)$ – распределенная (погонная) внешняя поперечная нагрузка; ось Ox направлена вдоль оси стержня.

Для решения уравнений (1.21) и (1.22) необходимо задать функции жесткости и распределенной внешней нагрузки в виде простых функций, чтобы иметь возможность получить аналитическое решение в приемлемом виде. Для этого нагрузку аппроксимируют постоянной функцией: $p(x) \cong p = const$, $q(x) \cong q = const$, а функцию жесткости **специальной степенной функцией**, представляющей собой многочлен Лагранжа первой степени, далее возведенный в некоторую степень:

$$\begin{aligned}
 C(x) &\cong B^\gamma(x) = \{[(b-x)B(a) + (x-a)B(b)]/(b-a)\}^\gamma, \\
 B(a) &= \sqrt[\gamma]{C(a)}, \quad B(b) = \sqrt[\gamma]{C(b)}; \quad (1.23) \\
 EI(x) &\cong e^\gamma(x) = \{[(b-x)e(a) + (x-a)e(b)]/(b-a)\}^\gamma, \\
 e(a) &= \sqrt[\gamma]{EI(a)}, \quad e(b) = \sqrt[\gamma]{EI(b)}.
 \end{aligned}$$

С учетом аппроксимации (1.23) уравнения (1.21) и (1.22) переписуются в следующем виде:

$$- [B^\gamma(x)u'(x)]' = p \quad \text{и} \quad [e^\gamma(x)v''(x)]'' = q \quad \forall x \in (a, b). \quad (1.24)$$

Далее для нахождения решений уравнений (1.24) их нужно проинтегрировать. Для этого используют **подстановку специального вида**:

$$\begin{aligned}
 \frac{d[B(x)]}{dx} = k = [B(b) - B(a)]/(b-a) &\Rightarrow dx = \frac{d[B(x)]}{k}; \\
 dx = \frac{d[e(x)]}{k}. & \quad (1.25)
 \end{aligned}$$

Будем последовательно понижать порядок уравнений (1.24) операцией неопределенного интегрирования с использованием подстановки (1.25). Первое уравнение для растяжения-сжатия и кручения проинтегрируем один раз, второе для изгиба – два раза. После того как избавимся от скобок, функции жесткости перенесем в правую часть, получим:

$$\begin{aligned}
 u'(x) &= -\frac{pB^{1-\gamma}(x)}{k} + \frac{c_1}{B^\gamma(x)} \\
 \text{и } v''(x) &= q\frac{e^{2-\gamma}(x)}{2k^2} + c_1\frac{e^{1-\gamma}(x)}{k} + c_2\frac{1}{e^\gamma(x)}, \quad \forall x \in (a, b), \quad (1.26)
 \end{aligned}$$

где c_1, c_2 – константы интегрирования, получаемые после первого и второго интегрирования соответственно и определяемые из краевых условий уравнений (условий закрепления, опирания или нагружения сечений стержня).

Дальнейшее интегрирование выражений зависит от показателя степени γ . Для растяжения-сжатия и кручения интеграл может быть получен для четырех случаев: $\gamma=0$, $\gamma=1$, $\gamma=2$ и

$\forall \gamma \neq 0, 1, 2$; при изгибе – для шести случаев: $\gamma = 0$, $\gamma = 1$, $\gamma = 2$, $\gamma = 3$, $\gamma = 4$ и $\forall \gamma \neq 0, 1, 2, 3, 4$.

Возникает задача определения показателя степени γ , для отыскания которого необходимо сформулировать три условия, так как в выражении (1.23) имеется три неизвестных $B(a)$, $B(b)$ и γ или $e(a)$, $e(b)$ и γ (далее рассмотрим только $B(a)$, $B(b)$ и γ , так как эти задачи аналогичны). Два условия уже сформулированы – это условия интерполирования: $B(a) = \sqrt[\gamma]{C(a)}$ и $B(b) = \sqrt[\gamma]{C(b)}$. Необходимо сформулировать третье условие, и здесь известны два варианта.

В первом варианте формулируется третье условие интерполирования в середине рассматриваемого участка:

$$B\left(\frac{b-a}{2}\right) \equiv \frac{B(a)+B(b)}{2} = \sqrt[\gamma]{C\left(\frac{b-a}{2}\right)} \equiv \sqrt[\gamma]{C_{cp}},$$

подставим в это условие первые два условия интерполирования и перенесем все в левую часть, получим нелинейное уравнение:

$$\frac{\sqrt[\gamma]{C(a)} + \sqrt[\gamma]{C(b)}}{2} - \sqrt[\gamma]{C_{cp}} = 0. \quad (1.27)$$

Таким образом, задача аппроксимации функции жесткости свелась к нелинейному уравнению (1.27), корни которого можно найти графическим способом. Для численного решения можно применить метод простой итерации (последовательных приближений). Здесь необходимо из уравнения (1.27) выразить показатель степени γ , но это приведет к тому, что итерации будут расходиться, поэтому уравнение (1.27) приходится преобразовывать к такому виду, чтобы итерации сходились. Рассмотрим эти преобразования на данном примере. Уравнение (1.27) преобразуется к виду:

$$\left(\frac{C(a)}{C_{cp}}\right)^{\frac{1}{\gamma}} + \left(\frac{C(b)}{C_{cp}}\right)^{\frac{1}{\gamma}} = 2;$$

затем производится замена к виду:

$$n^x + m^x = 2,$$

где $n = C(a)/C_{cp}$, $m = C(b)/C_{cp}$, $x = 1/\gamma$;

далее это уравнение можно привести к виду:

$$e^{x \ln(n)} + e^{x \ln(m)} = 2,$$

где e – основание натурального логарифма;

проведем еще одну замену и получим:

$$e^{cy} + e^y = 2,$$

где $c = \ln(n)/\ln(m)$, $y = x \ln(m)$;

используя еще одну замену, приведем уравнение к окончательному виду:

$$z^c + z = 2,$$

где $z = e^y$.

Итерационная формула нахождения корней уравнения получится выражением неизвестной z через оставшиеся члены уравнения:

$$z_{(i)} = 2 - z_{(i-1)}^c,$$

где (i) – порядковый номер итерации (приближения).

В начальном приближении берется $z_{(0)} > 1$, то есть строго больше одного из корней уравнения, если он не подходит из условий задачи. Определяются $z_{(i)}$ последующих итераций до тех пор, пока не выполнится условие сходимости итераций:

$$|z_{(i)} - z_{(i-1)}| \leq \varepsilon,$$

где ε – заданное малое число (точность).

Определив z , вычисляют показатель степени по формуле

$$\gamma = \frac{\ln(C(b)/C_{cp})}{\ln(z)}.$$

Во втором варианте формулируется условие равенства интегралов аппроксимирующей и аппроксимируемой функции:

$$\int_a^b B^\gamma(x) dx = \int_a^b C(x) dx \equiv I,$$

тогда уравнение можно записать как:

$$\int_a^b B^\gamma(x) dx - I = 0. \quad (1.28)$$

Для решения уравнения (1.28) необходимо получить выражения для интеграла аппроксимирующей функции с использо-

ванием специальной подстановки (1.25) и подставить в него первые два условия интерполирования:

$$\begin{aligned} \int_a^b B^\gamma(x) dx &= \int_{B(a)}^{B(b)} B^\gamma(x) \frac{d[B(x)]}{k} = \frac{B^{\gamma+1}(b) - B^{\gamma+1}(a)}{k(\gamma+1)} = \\ &= \frac{(b-a)[B^{\gamma+1}(b) - B^{\gamma+1}(a)]}{(\gamma+1)[B(b) - B(a)]} = \\ &= \left[(b-a) \left(C^{\frac{\gamma+1}{\gamma}}(b) - C^{\frac{\gamma+1}{\gamma}}(a) \right) \right] / \left[(\gamma+1) \left(C^{\frac{1}{\gamma}}(b) - C^{\frac{1}{\gamma}}(a) \right) \right]. \end{aligned} \quad (1.29)$$

Подставляя выражение для интеграла (1.29) в уравнение (1.28), получим нелинейное алгебраическое уравнение для нахождения γ :

$$\left[(b-a) \left(C^{\frac{\gamma+1}{\gamma}}(b) - C^{\frac{\gamma+1}{\gamma}}(a) \right) \right] / \left[(\gamma+1) \left(C^{\frac{1}{\gamma}}(b) - C^{\frac{1}{\gamma}}(a) \right) \right] - I = 0. \quad (1.30)$$

Итерационную формулу получим, выразив γ из уравнения (1.30):

$$\gamma_{(i)} = \left[(b-a) \left(C^{\frac{\gamma_{(i-1)}+1}{\gamma_{(i-1)}}}(b) - C^{\frac{\gamma_{(i-1)}+1}{\gamma_{(i-1)}}}(a) \right) \right] / \left[I \cdot \left(C^{\frac{1}{\gamma_{(i-1)}}}(b) - C^{\frac{1}{\gamma_{(i-1)}}}(a) \right) \right] - 1. \quad (1.31)$$

1.7. Основные понятия и определения векторно-матричного исчисления

Основным понятием векторно-матричного исчисления является вектор. Тем не менее, вычислительные машины изначально оперируют не с векторами, а с переменными и массивами, поэтому начнем с них. **Переменная** обладает именем и числовым значением, хранящимся в памяти ЭВМ. Однотипные переменные для удобства их вызова объединяются в **массивы**, обладающие одним именем и индексным обозначением переменных. Например, переменные $a = 2$, $b = 4$, $c = 7$ можно объединить в одномерный массив D , при этом элементами массива будут $D(1) = 2$, $D(2) = 4$, $D(3) = 7$. С переменными можно совершать любые арифметические действия, с массивами дейст-

вия производятся поэлементно. Массивы могут быть как одномерными, так и многомерными. Двумерные и многомерные массивы еще называют *матрицами*, однако операции над ними также проводятся поэлементно, а не по правилам векторно-матричного исчисления.

Близким одномерному массиву по смыслу является *строка*, но на нее уже наделаются определенные правила, зависящие от программного обеспечения. В векторно-матричном исчислении используется понятие *вектор-строки* (строки), представляющей собой упорядоченный набор чисел или переменных, записанных в строчку, и обозначается как $\mathbf{x} = [x_0, x_1, \dots, x_n]$. *Вектор-столбец* (вектор) представляет собой упорядоченный набор чисел или переменных, записанных в столбец, и обозначается как:

$$\mathbf{y} = \begin{bmatrix} y_0 \\ y_1 \\ \dots \\ y_n \end{bmatrix}.$$

Для экономии места вектора иногда записывают в строчку, заключенную в фигурные скобки, $\mathbf{y} = \{y_0, y_1, \dots, y_n\}$. Вектор является направленным отрезком в обобщенном n -мерном евклидовом пространстве. Компоненты (элементы) вектора являются длинами его проекций на оси этого пространства. Поэтому на операции с векторами наделаются определенные правила.

Для того чтобы преобразовать строку в вектор, используется операция *транспонирования*:

$$[x_0, x_1, \dots, x_n]^T = \{x_0, x_1, \dots, x_n\}; \{x_0, x_1, \dots, x_n\}^T = [x_0, x_1, \dots, x_n].$$

Рассмотрим еще одну операцию – *скалярное умножение* векторов:

$$\mathbf{x}\mathbf{y} = x_0y_0 + x_1y_1 + \dots + x_ny_n = \sum_{i=0}^n x_iy_i.$$

Таким образом, скалярным произведением двух векторов является скалярная (числовая) величина. Чтобы определить длину вектора (*модуль*), необходимо его скалярно перемножить на себя же и взять от получившегося скаляра квадратный корень:

$$|\mathbf{x}| = \sqrt{\mathbf{x}\mathbf{x}} = \sqrt{\sum_{i=0}^n x_i^2} \quad (\text{теорема Пифагора}).$$

При перемножении строк и векторов имеет место **порядок умножения**, так, например, при умножении строки на вектор получится скаляр, а при умножении вектора на строку получится матрица. Рассмотрим умножение вектора на строку с одинаковыми элементами:

$$\mathbf{xx}^T = \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix} \cdot [x_0, x_1, \dots, x_n] = \begin{bmatrix} x_0x_0 & x_0x_1 & \dots & x_0x_n \\ x_0x_1 & x_1x_1 & \dots & x_1x_n \\ \dots & \dots & \dots & \dots \\ x_0x_n & x_1x_n & \dots & x_nx_n \end{bmatrix} = \mathbf{X}.$$

Таким образом, получилась квадратная симметричная (относительно главной диагонали) матрица. Данный прием часто используется в инженерных расчетах.

Данные, содержащиеся в векторах, часто приходится представлять в трехмерном физическом пространстве. В этом случае используют составной вектор:

$$\mathbf{r} = x\mathbf{i}_1 + y\mathbf{i}_2 + z\mathbf{i}_3,$$

где $\mathbf{i}_1 = \{1, 0, 0\}$, $\mathbf{i}_2 = \{0, 1, 0\}$, $\mathbf{i}_3 = \{0, 0, 1\}$ – орты (единичные вектора) координатных осей Ox , Oy , Oz соответственно, x , y , z – проекции вектора \mathbf{r} на оси координат, которые, в свою очередь, могут также быть векторами. Если начало вектора \mathbf{r} находится в начале системы координат, то его проекции (компоненты) являются координатами точки конца этого вектора. Такой вектор задает точки трехмерного пространства, его компоненты являются координатами этих точек, и он называется **радиус-вектором**.

Если проекции радиус-вектора x , y , z являются векторами с бесконечным числом компонентов, то речь может идти о **координатных функциях**, зависящих от некоторого параметра t :

$$x = f_1(t), \quad y = f_2(t), \quad z = f_3(t).$$

В этом случае можно говорить о **вектор-функции**:

$$\mathbf{r}(t) = f_1(t)\mathbf{i}_1 + f_2(t)\mathbf{i}_2 + f_3(t)\mathbf{i}_3,$$

являющейся уравнением кривой в параметрическом виде. В качестве примера можно рассмотреть параметрическое уравнение окружности на плоскости:

$$\mathbf{r}(t) = R \cos(t) \mathbf{i}_1 + R \sin(t) \mathbf{i}_2,$$

где R – радиус окружности; параметр t – угол наклона радиуса к оси Ox .

Отметим, что скалярным произведением вектор-функций будет интеграл их компонентов (бесконечная сумма):

$$\mathbf{r}(t) \cdot \mathbf{r}(t) = \int_{a=t_{\min}}^{b=t_{\max}} \mathbf{r}^2(t) dt.$$

Несмотря на удобство векторно-матричного исчисления, ЭВМ не всегда работают с векторами, а тем более с вектор-функциями, поэтому их необходимо переводить в скалярный вид. Вектор-функция может быть разложена на три скалярные координатные функции – проекции вектор-функции на оси, скалярным умножением на соответствующие орты. Например, для двумерного случая:

$$\begin{aligned} \mathbf{r}(t) \mathbf{i}_1 &= [f_1(t) \mathbf{i}_1 + f_2(t) \mathbf{i}_2] \mathbf{i}_1 = f_1(t) \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + f_2(t) \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \\ &= f_1(t) \cdot (1 \cdot 1 + 0 \cdot 0) + f_2(t) \cdot (0 \cdot 1 + 1 \cdot 0) = f_1(t). \end{aligned}$$

Векторно-матричное исчисление реализовано в математическом программном обеспечении. Здесь все переменные представляются двумерными матрицами размерности 1×1 . Вектора и строки тоже реализованы как частные случаи матрицы. Поэтому операции над переменными, строками и векторами выполняются по умолчанию по правилам векторно-матричного исчисления. При необходимости выполнения поэлементных операций необходимо дополнительно это указать в операторе арифметического действия, например в MATLAB перед оператором устанавливается точка, означающее поэлементное выполнение операции.

1.8. Вопросы для самопроверки

1. Что такое силовая конструкция?
2. Какими параметрами описывается силовая конструкция?
3. Что включают в себя проектные параметры?
4. Что включают в себя параметры нагрузок?
5. Что включают в себя параметры перемещений?
6. На какие формы разделяются инженерные задачи?
7. На какие постановки разделяются инженерные задачи?
8. В какой постановке решается проблема задания геометрии конструкции?
9. Что такое численные методы?
10. Какие виды математических задач решают численные методы?
11. В чем заключается стратегия численных методов?
12. Каким методом решается система линейных алгебраических уравнений с нижней треугольной матрицей?
13. Каким методом решается система линейных алгебраических уравнений с верхней треугольной матрицей?
14. Что является целью численных методов?
15. Какие виды прикладного программного обеспечения реализуют алгоритмы численных методов?
16. Какие преимущества и недостатки у языков программирования высокого уровня?
17. Какие преимущества и недостатки у вычислительных сред?
18. Какие преимущества и недостатки у сред символьной математики?
19. Какие преимущества и недостатки у систем автоматизации инженерных расчетов?
20. Какие методы решения систем линейных алгебраических расчетов применяются в задачах анализа и проектирования автомобильных конструкций?
21. В чем заключается алгоритм метода Гаусса?
22. Что такое выбор ведущего (главного) элемента?
23. В чем заключается алгоритм метода алгебраической прогонки и для каких систем он применим?

24. *Что такое аппроксимация?*
25. *Что такое интерполяция?*
26. *Какие преимущества и недостатки у многочлена Лагранжа?*
27. *Какие преимущества и недостатки у многочлена Ньютона?*
28. *К каким задачам сводятся численное дифференцирование и интегрирование?*
29. *Как выводятся формулы конечной разности и трапеции?*
30. *Как выводится формула Симпсона?*
31. *Как выводится формула Симпсона $\frac{3}{8}$?*
32. *Как выводится формула Буля?*
33. *Как используются численные методы для получения аналитических решений дифференциальных уравнений деформации конструкций?*
34. *Что представляет собой специальная степенная функция?*
35. *Какая подстановка специального вида используется для дифференцирования и интегрирования специальной степенной функции?*
36. *Какие имеются варианты для определения показателя степени специальной степенной функции?*
37. *Что такое переменная и массив, строка и матрица?*
38. *Что такое вектор-строка, вектор-столбец и матрица?*
39. *Какие операции применяются над векторами и матрицами?*
40. *Что такое радиус-вектор, координатные функции и вектор-функция?*
41. *Как разложить вектор-функцию на скалярные координатные функции?*

ГЛАВА II. ПРОЧНОСТНОЙ АНАЛИЗ АВТОМОБИЛЬНЫХ КОНСТРУКЦИЙ

2.1. Численные методы анализа

Под задачей инженерного *анализа* конструкции понимают нахождение компонентов напряженно-деформированного состояния этой конструкции. То есть результатом решения этой задачи будут вектора напряжений и деформаций, содержащие в качестве своих компонентов значения напряжений и деформаций в заданных точках конструкции, соответственно. При этом полагают заданными (фиксированными) вектора проектных параметров и внешних нагрузок. При решении задачи анализа в форме метода перемещений сначала отыскивается вектор перемещений, затем по нему несложно методами теории упругости найти остальные вектора деформаций и напряжений. В этом случае, зная вектор перемещений, можно говорить, что получено решение задачи анализа.

Для получения вектора или функции перемещений необходимо решать дифференциальное уравнение деформации конструкции, для которого сложно на практике получить аналитическое решение. Поэтому применяют численные методы анализа. Широко распространенным методом для решения дифференциальных уравнений деформации конструкции является *метод конечных элементов*, однако он является частным случаем *метода Рунца* и также входит в логическую последовательность (цепь) численных методов анализа. Рассмотрим эту последовательность методов для решения задачи анализа.

Итак, задача анализа сводится к решению дифференциального уравнения деформации конструкции, аналитическое решение которого в общем случае неизвестно. От уравнения деформации конструкции переходят к рассмотрению полной потенциальной энергии этой конструкции. Отметим, что *полная потенциальная энергия* равна потенциальной энергии за вычетом работы внешних нагрузок. Искомая функция перемещений отыскивается теперь не из решения дифференциального уравнения, а из условия минимума полной потенциальной энергии деформации конструкции. Принцип минимума энергии является од-

ним из общих и универсальных принципов естествознания. Аналитическое решение задачи о минимуме энергии (минимизации) также не известно, поэтому для ее решения применяют метод Ритца или его частный случай – метод конечных элементов. Метод Ритца предназначен для минимизации функционалов (в данном случае **функционал** – это функция, аргументом которой выступает другая функция), к которым относится и выражение полной потенциальной энергии как функция от функции перемещений. Метод Ритца преобразует функционал в **функцию-форму** от конечного числа параметров (значений функции – аргумента функционала). То есть функция перемещений с бесконечным числом своих значений преобразуется в вектор с конечным числом компонентов – значений этой функции перемещений в заданных точках, а функционал полной потенциальной энергии преобразуется в функцию-форму, аргументами которой уже будут выступать это конечное число компонентов вектора перемещений. Следовательно, метод Ритца (или метод конечных элементов) сводит задачу минимизации функционала к задаче **минимизации функции** нескольких аргументов. Для отыскания минимума этой функции необходимо найти ее частные производные по всем компонентам вектора перемещений и приравнять их нулю, таким образом, задача минимизации функции сведется к системе дифференциальных уравнений. Далее необходимо получить выражения для частных производных минимизируемой функции и подставить их в эту систему уравнений – условий минимума. В задачах анализа это, как правило, приводит к системе линейных алгебраических уравнений, которая решается методом Гаусса для разреженных матриц с выбором ведущего элемента столбца.

Таким образом, **логическая последовательность численных методов анализа** следующая: задача анализа заключается в решении дифференциального уравнения, от которого переходят к задаче минимизации функционала полной потенциальной энергии, которая с помощью метода Ритца сводится к задаче минимизации функции, а затем к системе дифференциальных уравнений, сводимых к системе линейных алгебраических уравнений, решаемой методом Гаусса для разреженных матриц с выбором ведущего элемента столбца. В результате получим век-

тор искомым перемещений, по которому возможно восстановить и функцию перемещений – это задача интерполяции.

При такой последовательности методов на каждом шаге также может появиться необходимость использования уже ранее рассмотренных простейших численных методов. Например, при задании исходных данных необходимо аппроксимировать функции жесткости и внешней нагрузки, а в методе Рунге могут понадобиться некоторые решения дифференциальных уравнений деформации простых конструкций в соответствии с аппроксимацией исходных данных.

Знание последовательности численных методов еще не дает возможность решить задачу анализа. Эту последовательность необходимо реализовать для каждого типа задач, а для этого необходимы дополнительные знания из *теории операторов* или *функционального анализа, теории обобщенных решений* операторных уравнений, умение владеть вычислительными приемами типа интегрирования по частям и освоение в совершенстве каждого из перечисленных методов. Далее рассмотрим эти вопросы по отдельности.

2.2. Основные понятия математического и функционального анализа. Операторные линейные уравнения и математическая теория их обобщенных решений

Основным понятием математического анализа является *функция* $y = f(x)$. Функция – это закон, который ставит в соответствие элементу или числу x элемент или число y . Современная математика уже давно не оперирует понятием функции, вместо которого использует более общее понятие *отображения* $y \xleftarrow{f} x$, здесь говорят, что элемент или группа элементов x множества X отобразились посредством закона f на множество Y в элемент или группу элементов y , принадлежащих этому множеству Y . Тем не менее, понятие функции традиционно осталось в математическом анализе и некоторых других отраслях математики, где под функцией понимают вещественную функцию, то есть, подставив в качестве аргумента функции x веще-

ственное число, получим значение функции – вещественное число y . Функция – это частный случай отображения. Отметим, что функции бывают одного аргумента (переменного) $y = f(x)$ и нескольких аргументов (переменных) $y = f(x_0, x_1, \dots, x_n)$.

Функции с бесконечным числом аргументов $y = f(x_0, x_1, \dots)$ называются **функционалами**, их можно представить, например, как функцию от другой функции $y = f(g(x))$. Изучением функционалов занимается функциональный или операторный анализ.

Основным понятием функционального (операторного) анализа является другой частный случай отображения – **оператор**. Впрочем, понятие оператора обобщает понятие функции. Оператор может поставить в соответствие одной функции другую функцию, например, оператор дифференцирования $D[f(x)] = f'(x)$ или $Df = f'$. Понятие оператора очень важно в математике и, в том числе, численных методах, так как позволяет значительно сократить и обобщить математические выкладки. Например, рассмотренные дифференциальные уравнения деформации конструкции можно представить в одном общем виде:

$$Lu = p,$$

где L – оператор, u – вектор функций перемещений, p – вектор функций внешних нагрузок.

Для уравнения деформации балки оператор L сначала дважды дифференцирует функцию прогибов, затем умножает на функцию изгибной жесткости, потом дважды дифференцирует это произведение. Фактически запись Lu заменяет левую часть дифференциального уравнения, а p – правую.

Тем не менее, операторная запись уравнения не позволяет сходу найти решение этого уравнения. Поэтому применяют **математическую теорию обобщенных решений линейных операторных уравнений**, предложенную С.Г. Михлиным. У большинства уравнений механики оператор L – линейный симметричный положительно определенный. Линейность оператора вытекает из линейной зависимости между напряжениями и деформациями (закон Гука), симметричность – из взаимности

энергий и работ, положительная определенность – из того, что потенциальная энергия деформации всегда положительная величина. Если задача нелинейная, то ее сводят к серии линейных задач. **Обобщенным решением** называется решение из области определения некоторого функционала, доставляющее минимум этому функционалу (реализующее стационарную точку этого функционала). При этом обобщенное решение совпадает с точным аналитическим (классическим) решением операторного уравнения. То есть классическое решение обобщается на область определения данного функционала. Задача получения функционала по исходному операторному уравнению неоднозначна. Наиболее удобен следующий вид функционала:

$$F(\mathbf{u}) = (L\mathbf{u}, \mathbf{u}) - 2(\mathbf{p}, \mathbf{u}),$$

где (\mathbf{x}, \mathbf{y}) – обозначение скалярного произведения элементов \mathbf{x} и \mathbf{y} .

Физический смысл этого функционала – удвоенная полная потенциальная энергия деформации конструкции.

Рассмотрим получение полной потенциальной энергии деформации стержня, работающего на растяжение-сжатие и описываемого уравнением:

$$- [EA(x)u'(x)]' = r(x), \quad \forall x \in (a, b).$$

Для него функционал запишется следующим образом:

$$\begin{aligned} F(u) &= (Lu, u) - 2(r, u) = \left(- [EA(x)u'(x)]', u(x) \right) - 2(r(x), u(x)) = \\ &= - \int_a^b [EA(x)u'(x)]' u(x) dx - 2 \int_a^b r(x)u(x) dx. \end{aligned}$$

В этом выражении использована формула скалярного умножения функций. Видно, что получилось два члена: первый – удвоенная потенциальная энергия и второй – удвоенная работа внешних сил. Преобразуем первый член – потенциальную энергию деформации стержня, используя **формулу интегрирования по частям**:

$$\int_a^b U dV = UV \Big|_a^b - \int_a^b V dU.$$

Положим:

$$U = u(x), \quad dV = [EA(x)u'(x)]' dx = d[EA(x)u'(x)],$$

тогда:

$$V = EA(x)u'(x), \quad dU = du(x) = u'(x)dx,$$

получим:

$$\begin{aligned} - \int_a^b [EA(x)u'(x)]' u(x) dx &= -[EA(b)u'(b)u(b) - EA(a)u'(a)u(a)] + \\ &+ \int_a^b EA(x)(u'(x))^2 dx. \end{aligned}$$

После преобразований функционал запишется как:

$$\begin{aligned} F(u) &= 2 \left[\frac{1}{2} \int_a^b EA(x)(u'(x))^2 dx - \int_a^b r(x)u(x) dx \right] - \\ &- [EA(b)u'(b)u(b) - EA(a)u'(a)u(a)]. \end{aligned}$$

В первой квадратной скобке находится полная потенциальная энергия деформации стержня, работающего на растяжение-сжатие. Во второй квадратной скобке – краевые условия стержня. Рассмотрим универсальные краевые условия – здесь их возможно всего три:

– стержень закреплен в обоих концах, то есть перемещения на концах равны нулю: $u(a) = 0$ и $u(b) = 0$;

– стержень закреплен в левом конце (при $x = a$), второй конец (правый, при $x = b$) свободен, причем к нему не приложена сосредоточенная осевая (растягивающая) сила:

$$N(x) = - \int r(x) dx = \int [EA(x)u'(x)]' dx = EA(x)u'(x),$$

значит $N(b) = EA(b)u'(b) = 0$, тогда условия запишутся: $u(a) = 0$ и $EA(b)u'(b) = 0$;

– стержень закреплен в правом конце и свободен в левом: $EA(a)u'(a) = 0$ и $u(b) = 0$.

Во всех трех случаях правая скобка равна нулю, и окончательное выражение функционала запишется как:

$$F(u) = 2 \left[\frac{1}{2} \int_a^b EA(x)(u'(x))^2 dx - \int_a^b r(x)u(x) dx \right].$$

2.3. Введение в вариационное исчисление и метод Ритца

Вариационное исчисление – это раздел математики, посвященный исследованию методов отыскания экстремумов (максимумов и минимумов) функционалов, зависящих от выбора функций. Численные методы, посвященные отысканию экстремальных значений функционалов, называют **численными методами вариационного исчисления**. Данные методы сводят задачу **минимизации функционала**, зависящего от бесконечного числа параметров, к задаче его минимизации для конечного числа параметров.

Современные численные методы минимизации функционалов основаны на методе Ритца. **Метод Ритца** заключается в том, что вместо искомой функции $u(x)$ в функционал $F(u)$ подставляется пробная функция в виде обобщенного многочлена:

$$u(x) \cong \varphi_0(x) + \sum_{i=1}^n a_i \varphi_i(x),$$

где $\varphi_i(x)$ – базисные функции из области определения функционала, a_i – неизвестные коэффициенты (параметры).

После подстановки пробной функции в функционал он становится функцией-формой $F_1(a_1, a_2, \dots, a_n)$ от неизвестных параметров a_i . Далее остается минимизировать эту форму по неизвестным параметрам, то есть найти частные производные этой формы по неизвестным параметрам и приравнять их к нулю:

$$\frac{\partial F_1}{\partial a_i} = 0, \quad i = 1, 2, \dots, n,$$

решить получившуюся систему уравнений, то есть найти неизвестные параметры a_i , затем подставить их в обобщенный многочлен, и приближенная искомая функция получена.

Сложность в методе Ритца вызывает вопрос о выборе пробной функции. **Пробная функция** должна соответствовать природе рассматриваемой задачи, обеспечивать необходимую степень свободы варьирования значениями (степень свободы возможных перемещений в задачах анализа). Наиболее желательно, чтобы пробная функция совпадала с решением диффе-

ренциального уравнения деформации конструкции. Тем не менее, возникает вопрос о том, как разложить решение на базисные функции. Рассмотрим его на примере растяжения-сжатия стержня постоянной жесткости, нагруженного распределенной внешней осевой нагрузкой постоянной интенсивности и описываемого уравнением:

$$-EAu''(x) = r, \quad \forall x \in (a, b).$$

Получим его решение, для чего разделим обе части на $-EA$ и дважды проинтегрируем:

$$u(x) = -\iint \frac{r}{EA} dx dx = -\frac{rx^2}{2EA} + c_1x + c_2,$$

где c_1 и c_2 – неизвестные константы интегрирования.

Теперь его нужно разложить на **базисные функции**, для этого определим неизвестные константы интегрирования через **краевые условия Эрмита** (в данном случае они совпадают с условиями интерполирования):

$$\begin{cases} u(a) = -\frac{ra^2}{2EA} + c_1a + c_2 \\ u(b) = -\frac{rb^2}{2EA} + c_1b + c_2 \end{cases}.$$

Определим из данной системы неизвестные константы интегрирования c_1 и c_2 , подставим их в решение, сгруппируем множители относительно $u(a)$ и $u(b)$, получим:

$$u(x) = -\frac{r(x-a)(x-b)}{2EA} - \frac{x-b}{b-a}u(a) + \frac{x-a}{b-a}u(b),$$

где обозначим базисные функции и неизвестные коэффициенты:

$$-\frac{r(x-a)(x-b)}{2EA} = \varphi_0(x), \quad -\frac{x-b}{b-a} = \varphi_1(x), \quad \frac{x-a}{b-a} = \varphi_2(x),$$

$$a_1 = u(a), \quad a_2 = u(b).$$

Тогда решение запишется в виде:

$$u(x) = \varphi_0(x) + \varphi_1(x)a_1 + \varphi_2(x)a_2 = \varphi_0(x) + \sum_{i=1}^2 \varphi_i(x)a_i.$$

Таким образом, типовое решение уравнения деформации разложено на базисные функции, которые можно применять для приближенной минимизации аналогичных, но более сложных

конструкций. Базисные функции $\varphi_i(x)$ показывают деформацию закрепленного стержня, получившего смещение в i -м направлении.

Для пробной функции можно использовать и однородные решения дифференциальных уравнений. В этом случае $\varphi_0(x) = 0$, а все неизвестные параметры a_i сохраняются. Для рассмотренного стержня однородное решение (правая часть уравнения – нагрузка $r = 0$) $u(x) = -\frac{x-b}{b-a}u(a) + \frac{x-a}{b-a}u(b)$ совпадает с полиномом Лагранжа первой степени (1.11).

Для балки – стержня, работающего на поперечный изгиб, число неизвестных параметров a_i удваивается и равно четырем. Это прогибы и их производные (тангенсы углов наклона касательной к линии прогибов): $a_1 = v(a)$, $a_2 = v'(a)$, $a_3 = v(b)$, $a_4 = v'(b)$.

Многочлен, разложенный таким способом на базисные функции, то есть с применением краевых условий Эрмита (условий интерполирования и опирания на производные функции), называется *эрмитовым многочленом*, а базисные функции в этом случае называются *функциями Эрмита*.

При решении сложных задач необходимо аппроксимировать пробной функцией сложную функцию решения, что с необходимой точностью можно выполнить лишь на некотором ограниченном участке. Поэтому используют аппроксимацию по участкам, на которые разбивают по значениям каких-либо параметров, например по значениям координат, составляющих сетку параметров. В этом случае методы решения называются *сеточными*.

2.4. Основные положения метода конечных элементов

Метод конечных элементов (МКЭ) является основным промышленным методом анализа напряженно-деформированного состояния элементов автомобильных конструкций, которые описываются дифференциальными уравнениями их деформации. Таким образом, МКЭ применяется для ре-

шения дифференциальных (операторных) уравнений деформации конструкций и основан на методе Рунге. МКЭ – это сеточный метод.

Конструкция описывается уравнением деформации в операторном виде:

$$L\mathbf{u} = \mathbf{p}, \quad (2.1)$$

где $L = L(\mathbf{r})$ – оператор, \mathbf{r} – вектор проектных параметров (вектор функций жесткости), \mathbf{u} – вектор функций перемещений, \mathbf{p} – вектор функций внешних нагрузок.

При задании конструкции необходимо аппроксимировать функцию жесткости и функцию внешней распределенной нагрузки. Для этого используют кусочные полиномы. В этом случае конструкция разбивается на конечные элементы (кусочные фрагменты, участки), значения функций жесткости и внешней нагрузки на которых изменяются по полиномиальным законам или постоянны. Конечные элементы состыкованы между собой условиями сопряжения.

Уравнению (2.1) соответствует функционал:

$$F(\mathbf{u}) = (L\mathbf{u}, \mathbf{u}) - 2(\mathbf{p}, \mathbf{u}). \quad (2.2)$$

Обобщенное решение \mathbf{u} доставляет минимум функционалу (2.2) и совпадает с решением уравнения (2.1).

Для минимизации функционала (2.2) на сетке параметров $\Delta: a = x_0 < x_1 < \dots < x_n = b$ в качестве пробных функций принимаются локальные финитные (конечные) многочлены:

$$u(x) = \sum_{i=1}^k a_{i,j} \varphi_{i,j}(x), \quad \forall x \in [x_{j-1}, x_j], \quad (2.3)$$

где k – количество неизвестных коэффициентов функции перемещений на конечном элементе (КЭ); j – порядковый номер КЭ.

Пробная функция (2.3) конструируется путем разложения на базисные функции однородного решения уравнения (2.1) $L\mathbf{u} = 0$ и может быть представлена как линейная форма в векторно-матричном виде:

$$u(x) = \mathbf{a}_j^T \{\varphi_i\}_j = \{\varphi_i\}_j^T \mathbf{a}_j, \quad (2.4)$$

где \mathbf{a}_i – вектор неизвестных коэффициентов пробной функции – вектор неизвестных узловых перемещений j -го конечного элемента; $\{\varphi_i\}_j$ – вектор базисных функций j -го конечного элемента.

Функции жесткости и внешней нагрузки аппроксимированы аналогично. После подстановки их и пробной функции (2.4) в функционал (2.2) из него получается квадратичная форма, зависящая от неизвестных коэффициентов пробной функции:

$$F(\mathbf{a}) = \sum_{i=1}^n \mathbf{a}_i^T \mathbf{K}_i \mathbf{a}_i - 2 \sum_{i=1}^n \mathbf{a}_i^T \mathbf{p}_i, \quad (2.5)$$

где \mathbf{K}_i – матрица жесткости i -го конечного элемента; \mathbf{p}_i – вектор эквивалентных узловых нагрузок i -го конечного элемента.

Далее записываются условия минимума квадратичной формы (2.5) по искомым коэффициентам \mathbf{a} :

$$\frac{\partial F(\mathbf{a})}{\partial \mathbf{a}^T} = 0,$$

из которых следует система линейных алгебраических уравнений – зависимость между узловыми усилиями и перемещениями конструкции:

$$\mathbf{K}\mathbf{a} = \mathbf{p}. \quad (2.6)$$

Система уравнений (2.6) собрана для всех конечных элементов конструкции. Матрица системы – матрица жесткости является вырожденной (особенной), ее определитель равен нулю. В этом случае возможно бесконечное число решений, то есть конструкция перемещается в пространстве как абсолютно жесткое тело. Чтобы получить единственное решение этой системы, в ней необходимо учесть краевые условия задачи – условия закрепления конструкции. Условие закрепления записывается уравнением $a_i = 0$, на которое необходимо заменить соответствующее уравнение системы (2.6), где i – порядковый номер коэффициента, соответствующего закрепленному узлу.

Таким образом, определив из системы (2.6) совместно с краевыми условиями неизвестные коэффициенты \mathbf{a} и подставив их в пробную функцию (2.3), для каждого конечного элемента строится функция перемещений, зная которую, можно перейти к напряжениям методами сопротивления материалов.

Следует отметить, что МКЭ может быть представлен в форме как метода перемещений, так и метода сил. В зависимости от решаемой задачи используются различные конечные элементы: одномерные (стержневые), двумерные (пластинчатые, оболочечные), объемные (твердотельные). КЭ могут иметь прямолинейные или криволинейные образующие (оси, стороны, ребра). Двумерные (плоские) КЭ могут быть треугольными или четырехугольными. Объемные КЭ могут быть в виде тетраэдра, клина или гексаэдра.

Также в зависимости от способов аппроксимации исходных данных задачи – функции жесткости и функции внешней нагрузки, учета деформации, воспринимаемой конструкцией, имеется множество различных видов конечных элементов.

Правильность решения задачи анализа конструкции не гарантируется формальным применением МКЭ. Необходимо грамотно построить расчетную модель конструкции, а затем подобрать подходящие типы конечных элементов, то есть построить расчетную конечно-элементную модель. Универсализация конечно-элементных моделей приводит к весьма завышенным вычислительным затратам, так например, для анализа балки может быть использовано несколько (не более 10 – 20) балочных КЭ, а в универсальных моделях пакетов конечно-элементного анализа балка может быть рассмотрена как объемное твердое тело с привлечением тысяч объемных КЭ типа гексаэдров. Такая универсализация сделает невозможным анализ сложных конструкций. Поэтому для решения задач анализа конструкций с помощью МКЭ необходимо знать как механику конструкций, так и численные методы (в том числе и МКЭ), с целью построения расчетной модели, адекватно учитывающей особенности конструкции и вычислительные возможности.

2.5. Механическая аналогия между деформацией стержней и интерполяцией сплайнами

Задача интерполяции заключается в проведении гладкой кривой по точкам. Часто функции задаются таблицей значений, при этом и возникает задача нахождения значений этой функции, находящихся между ее заданными значениями в таблице. В

принципе, при построении плоской кривой на плоской доске можно расчертить систему координат, на ней отметить точками заданные значения функции и вбить туда гвозди (не до конца), к которым прижать металлическую линейку. Линейка гладко изогнется, огибая забитые в заданных точках гвозди. Функция прогибов этой линейки и будет решением задачи интерполяции.

Как же математически решить эту задачу, не прибегая к физическому эксперименту? В данном случае мы имеем **изгиб неразрезной многоопорной балки постоянной жесткости**, функцию прогибов которой можно найти МКЭ. Поставим задачу следующим образом: необходимо найти функцию прогибов неразрезной многоопорной балки постоянной жесткости под действием внутренних реакций, возникших в результате заданного смещения опор балки. Такая задача называется **многоточечной краевой задачей**. Отметим, что внешняя нагрузка к балке не прикладывается, поэтому необходимо решить однородное дифференциальное уравнение деформации балки:

$$[EI(x)v''(x)]'' = 0, EI(x) = const \Rightarrow v^{IV}(x) = 0 \quad \forall x \in (a, b). \quad (2.7)$$

Чтобы учесть заданные перемещения опор, необходимо использовать сеточные методы решения уравнений. Будем решать уравнение (2.7) на заданной сетке узлов $\Delta: a = x_0 < x_1 < \dots < x_n = b$ по участкам:

$$v^{IV}(x) \equiv s^{IV}(x) = 0 \quad \forall x \in (x_{i-1}, x_i), i = 1, 2, \dots, n. \quad (2.8)$$

Решением уравнения (2.8) будет полином третьей степени с неизвестными коэффициентами:

$$s(x) = \iiint \int 0 dx dx dx dx = c_{1,i}x^3 + c_{2,i}x^2 + c_{3,i}x + c_{4,i} \quad \forall x \in (x_{i-1}, x_i), i = 1, 2, \dots, n. \quad (2.9)$$

Разложим решение (2.9) на базисные функции через крайвые (в данном случае, граничные) условия Эрмита:

- 1) $s(x_{i-1}) = s_{i-1}$ и $s(x_i) = s_i$ (условия интерполирования);
- 2) $s'(x_{i-1}) = s'_{i-1}$ и $s'(x_i) = s'_i$ (условия опирания),

где s_{i-1}, s_i – заданные смещения опор балки; s'_{i-1}, s'_i – неизвестные коэффициенты – производные функции прогибов;

получим:

$$s(x) = \varphi_{1,i}(x)s_{i-1} + \varphi_{2,i}(x)s'_{i-1} + \varphi_{3,i}(x)s_i + \varphi_{4,i}(x)s'_i, \\ \forall x \in [x_{i-1}, x_i], \quad i = 1, 2, \dots, n. \quad (2.10)$$

Неизвестные значения производной функции прогибов в узлах найдем из условия непрерывности функции изгибающего момента $M(x) = EI(x)v''(x)$:

$$s''(x_j - 0) = s''(x_j + 0), \quad j = 1, 2, \dots, n - 1. \quad (2.11)$$

Системе (2.11) не хватает двух уравнений, так как на краях балки нет сопряжения участков. Два уравнения берутся из крайних условий, которых имеется четыре типа:

1) на краях задаются значения первых производных (касательные) функции прогибов: $s'(x_j) = s'_j, \quad j = 0, n$;

2) на краях задаются значения вторых производных (кривизна) функции прогибов: $s''(x_j) = s''_j, \quad j = 0, n$;

3) для замкнутых или периодических кривых имеется сопряжение первого и последнего участка: $s''(x_0) = s''(x_n)$;

4) когда не известны ни первые, ни вторые производные на краях и кривая не является периодической, тогда предполагают неразрывность третьей производной на парах крайних участков: $s'''(x_j - 0) = s'''(x_j + 0), \quad j = 1, n - 1$.

Решив систему (2.11) совместно с крайевыми условиями, найденные узловые значения производной функции прогибов подставляются в выражение для участка (2.10). По участкам строится интерполирующая функция, называемая **кубическим сплайном**.

Для того чтобы уменьшать или увеличивать прогибы балки, при такой интерполяции необходимо соответственно увеличивать или уменьшать функцию жесткости. В этом случае кубический сплайн неприменим. Необходимо строить сплайны, аналогичные балкам переменной жесткости.

Сплайны переменной жесткости строятся аналогично. Решается однородное уравнение изгиба балки переменной жесткости $[e(x)s''(x)]'' = 0$, решение которого раскладывается на базисные функции через крайевые условия Эрмита. Неизвестные

коэффициенты сплайна находятся из условий непрерывности функции изгибающего момента:

$$e(x_j - 0)s''(x_j - 0) = e(x_j + 0)s''(x_j + 0), \quad j = 1, 2, \dots, n - 1.$$

Недостающие два уравнения берутся из краевых условий:

$$1) \quad s'(x_j) = s'_j, \quad j = 0, n;$$

$$2) \quad s''(x_j) = s''_j, \quad j = 0, n;$$

$$3) \quad e(x_0)s''(x_0) = e(x_n)s''(x_n);$$

$$4) \quad \text{в двух вариантах } s'''(x_j - 0) = s'''(x_j + 0), \quad j = 1, n - 1 \text{ или}$$

$$\left[e(x_j - 0)s''(x_j - 0) \right]' = \left[e(x_j + 0)s''(x_j + 0) \right]', \quad j = 1, n - 1.$$

Также возможно получение решения из функционала. Такие сплайны называются *функциональными*. Сплайновое операторное уравнение имеет вид: $LS = 0$, ему соответствует функционал $F(s) = (Ls, s)$. Подставляя в функционал решение сплайнового уравнения, разложенного на базисные функции (2.10), получаем квадратичную форму, зависящую от неизвестных коэффициентов сплайна. Далее решая задачу минимизации этой формы, получаем систему линейных алгебраических уравнений для нахождения неизвестных коэффициентов сплайна. В этом случае алгоритм построения сплайнов совпадает с алгоритмом МКЭ. Матрицу жесткости здесь называют матрицей Грама.

Следует отметить, что система уравнений для нахождения неизвестных коэффициентов сплайна имеет трехдиагональную матрицу и наилучшим образом решается методом алгебраической прогонки.

2.6. Задача параметризации и методы её решения

При решении задачи анализа конструкции необходимо задать ее геометрию, имеющую сложные формы, в аналитическом виде. То есть необходимо решать задачи аппроксимации или интерполяции, таким образом построить кривую, описывающую геометрию анализируемой конструкции.

При практическом построении кривых широко используют их **векторно-параметрическое уравнение**:

$$\mathbf{r}(t) = f_1(t)\mathbf{i}_1 + f_2(t)\mathbf{i}_2 + f_3(t)\mathbf{i}_3, \quad (2.12)$$

где $\mathbf{r}(t)$ – радиус-вектор – вектор-функция от параметра t ; $f_i(t)$ – координатные функции; \mathbf{i}_i – орты (единичные вектора) координатных осей; $i = 1, 2, 3$.

Векторное уравнение (2.12) эквивалентно трем **скалярным уравнениям**:

$$x = f_1(t), \quad y = f_2(t), \quad z = f_3(t), \quad (2.13)$$

где x, y, z – проекции радиус-вектора $\mathbf{r}(t)$ на оси системы координат $Oxyz$ (координаты точек кривой).

Примером может служить ранее рассмотренное параметрическое уравнение окружности (раздел 1.7, с. 39).

Чтобы не рассматривать каждое из трех скалярных уравнений по отдельности, введем новые обозначения. Возьмем систему координат $Oy_1y_2y_3$, тогда точка кривой в данной системе будет задана координатами (y_1, y_2, y_3) . Параметр обозначим как x . В новых обозначениях векторно-параметрическое уравнение кривой (2.12) запишется как:

$$\mathbf{r}(x) = f_1(x)\mathbf{i}_1 + f_2(x)\mathbf{i}_2 + f_3(x)\mathbf{i}_3 = \sum_{i=1}^3 f_i(x)\mathbf{i}_i, \quad (2.14)$$

а эквивалентные скалярные уравнения в виде:

$$y_i = f_i(x), \quad i = 1, 2, 3.$$

Для построения кривой необходимо выбрать **сетку конечного числа значений параметров** $\Delta: a = x_0 < x_1 < \dots < x_n = b$, и, следовательно, определиться с параметром. К выбору параметра предъявляются следующие требования:

- 1) параметр должен однозначно определять точку кривой;
- 2) выражаться через известные функции;
- 3) обеспечивать одинаковый порядок следования точек по кривой.

Задача выбора подходящих значений сетки параметров называется **задачей параметризации**.

Известно множество практических способов выбора параметризации. Рассмотрим некоторые из них.

1. В качестве параметра выбрана одна из координат:

$$x = y_i.$$

Этот способ часто применяется для задания несложных кривых и построения их графиков. Он интуитивно понятен. Однако даже при задании замкнутой кривой на плоскости такой выбор параметра не соответствует первому требованию: одному значению параметра могут соответствовать сразу две и более точек на кривой.

2. Угловая параметризация. В качестве параметра выбирается угол наклона радиус-вектора к оси абсцисс:

$$x = y_2 / y_1.$$

Здесь в качестве параметра, собственно, также выбирается одна из координат, но уже полярной системы координат. Такой способ имеет более широкую область применимости и даже используется для задания окружности из примера. Однако имеет тот же существенный недостаток – не всегда удовлетворяет первому требованию.

3. Равномерная параметризация. Значения параметра сетки Δ берутся с равным шагом:

$$d_i = x_i - x_{i-1} = \text{const} = d, \quad i = 1, 2, \dots, n.$$

Отсюда значения параметра вычисляются по формуле:

$$x_0 = a, \quad x_i = x_{i-1} + d = a + id, \quad i = 1, 2, \dots, n, \quad b = x_n = a + nd.$$

В случае если $d = 1$, то значения параметра совпадают с порядковыми номерами точек кривой и упрощаются все расчетные выражения. Однако такой способ параметризации нельзя назвать удовлетворительным, так как он никак не зависит от геометрии кривой.

4. Натуральная (естественная) параметризация. Разумеется, самым лучшим способом параметризации будет такой, если в качестве параметра выбрать значение длины кривой от ее начала до рассматриваемой точки. За шаг сетки Δ выбирается длина кривой между двумя рассматриваемыми соседними точками x_{i-1} и x_i , которая вычисляется по формуле:

$$d_i = \int_{x_{i-1}}^{x_i} |\mathbf{r}'(x)| dx = \int_{x_{i-1}}^{x_i} \sqrt{(\mathbf{r}'(x))^2} dx,$$

тогда значения параметра вычисляются по формуле:

$$x_0 = a, \quad x_i = x_{i-1} + d_i = x_{i-1} + \int_{x_{i-1}}^{x_i} \sqrt{(\mathbf{r}'(x))^2} dx, \quad i = 1, 2, \dots, n, \quad b = x_n.$$

Однако до задания кривая неизвестна, поэтому такую задачу параметризации приходится решать методом последовательных приближений (итерационным методом), что требует большого количества вычислений. В начальном приближении строится кривая по любой из известных или ранее рассмотренных параметризаций, затем вычисляются длины участков этой кривой и значения сетки параметров. На новой сетке параметров строится следующая кривая, опять вычисляются длины участков этой кривой и так продолжают до тех пор, пока значения сетки параметров предыдущего приближения не сравняются с последующим приближением.

5. Параметризация по суммарной длине хорд (параметризация по хордам). Чтобы избежать большого объема вычислений, натуральную параметризацию упрощают. В качестве параметра выбирают длину ломаной линии, последовательно соединяющей прямыми отрезками все точки от начала кривой до рассматриваемой точки. За шаг сетки Δ выбирается длина прямого отрезка, соединяющего две рассматриваемые соседние точки x_{i-1} и x_i , которая вычисляется по формуле:

$$d_i = \sqrt{\sum_{i=1}^3 (y_i - y_{i-1})^2},$$

тогда значения параметра вычисляются по формуле:

$$x_0 = a, \quad x_i = x_{i-1} + d_i = x_{i-1} + \sqrt{\sum_{i=1}^3 (y_i - y_{i-1})^2}, \quad i = 1, 2, \dots, n, \quad b = x_n.$$

Такой метод широко применяется на практике и реализован в большинстве САПР.

Отметим, что при интерполировании параметрическими вектор-сплайнами $\mathbf{s}(x) \cong \mathbf{r}(x)$ **наилучшая параметризация** будет удовлетворять условиям минимума сплайнового функционала:

$$F(\mathbf{s}) = (L\mathbf{s}, \mathbf{s}),$$

которые запишутся в виде:

$$\frac{\partial F(\mathbf{s})}{\partial \mathbf{s}_i} = 0, \quad \frac{\partial F(\mathbf{s})}{\partial \mathbf{s}'_i} = 0, \quad \frac{\partial F(\mathbf{s})}{\partial d_j} = 0, \quad i = 0, 1, \dots, n, \quad j = 1, 2, \dots, n$$

с условием

$$\sum_{i=1}^n d_i - (b - a) = 0,$$

что составляет систему нелинейных алгебраических уравнений, решаемой методами оптимизации.

2.7. Вопросы для самопроверки

1. *Что понимается под задачей инженерного анализа?*
2. *В чем заключается логическая последовательность численных методов в задаче анализа?*
3. *Что такое функция и отображение?*
4. *Что такое функционал?*
5. *Что такое оператор?*
6. *В чем суть теории обобщенных решений линейных операторных уравнений?*
7. *Что такое обобщенное решение?*
8. *Как выводится функционал по исходному операторному уравнению?*
9. *Какой физический смысл имеет функционал в задачах анализа автомобильных конструкций?*
10. *Для чего используется интегрирование по частям при получении функционала?*
11. *Какие универсальные краевые условия подставляются в функционал?*
12. *Что такое вариационное исчисление?*
13. *Чему посвящены численные методы вариационного исчисления?*
14. *Как решается задача минимизации функционала?*
15. *В чем суть метода Рунца?*
16. *Что такое пробная функция?*
17. *Что такое базисные функции и как их получают?*
18. *В чем заключаются краевые условия Эрмита?*
19. *Что такое эрмитовый многочлен?*
20. *Что такое функции Эрмита?*
21. *Какие методы называются сеточными?*
22. *В чем суть алгоритма метода конечных элементов?*
23. *Как конструируется пробная функция в МКЭ?*

24. *Что такое матрица жесткости?*
25. *Что такое вектор эквивалентных узловых нагрузок?*
26. *Каким образом учитываются краевые условия в МКЭ?*
27. *Какие виды конечных элементов бывают?*
28. *В чем заключается механическая аналогия сплайнов с деформацией стержней?*
29. *В чем заключается многоточечная краевая задача?*
30. *Как решается изгиб неразрезной многоопорной балки постоянной жесткости?*
31. *В чем заключаются условия интерполирования?*
32. *В чем заключаются условия опирания?*
33. *Из каких условий отыскиваются неизвестные коэффициенты сплайна?*
34. *Какие краевые условия используются для сплайнов?*
35. *Что такое кубический сплайн?*
36. *Что такое сплайны переменной жесткости?*
37. *Что такое функциональные сплайны?*
38. *Что такое векторно-параметрическое уравнение кривой?*
39. *Каким образом векторно-параметрическое уравнение кривой эквивалентно трем скалярным уравнениям?*
40. *Что такое сетка параметров?*
41. *Какие требования предъявляются к выбору параметра?*
42. *Что такое задача параметризации?*
43. *Какие способы параметризации бывают?*
44. *Какие преимущества и недостатки у параметризации, где в качестве параметра выбирают одну из координат?*
45. *Какие преимущества и недостатки у угловой параметризации?*
46. *Какие преимущества и недостатки у равномерной параметризации?*
47. *Какие преимущества и недостатки у натуральной (естественной) параметризации?*
48. *Какие преимущества и недостатки у параметризации по суммарной длине хорд?*
49. *Как отыскивается наилучшая параметризация?*

ГЛАВА III. МЕТОДЫ ОПТИМАЛЬНОГО ПРОЕКТИРОВАНИЯ АВТОМОБИЛЬНЫХ КОНСТРУКЦИЙ

3.1. Численные методы оптимизации и минимизации для автомобильных конструкций

Перед конструкторами постоянно возникают задачи оптимизации конструкций по уменьшению массы и металлоемкости, повышению прочности и жесткости и тому подобные. Формально процесс оптимального проектирования заключается в том, что конструктор, анализируя оптимизируемую конструкцию, определяет картину (поля) ее напряженно-деформированного состояния и тем самым выявляет ее сильно- и слабонагруженные места. Далее конструктор перераспределяет материал силовой конструкции, перемещая его из слабонагруженных мест в сильнонагруженные. Затем опять проводится анализ конструкции, и процесс повторяется последовательными приближениями (итерациями) до тех пор, пока напряжения во всех точках силовой конструкции не сравняются. Затем одинаковые напряжения в конструкции сравнивают с допускаемыми (или разрушающими) напряжениями и соответственно (пропорционально) изменяют размеры (общую массу) конструкции.

На практике обычно такой процесс не доводят до конца, а довольствуются тем, что полученное конструктором промежуточное приближенное решение лучше по тем или иным насущным показателям, чем применявшееся ранее или применяющееся у конкурентов. Как говорится, нет предела совершенству, но рано или поздно нужно остановиться на достигнутом и выдать новый автомобиль.

Тем не менее, разработано множество методов решения оптимизационных задач, суть которых заключается в выборе функции, называемой *целевой функцией*, от оптимизируемых параметров. Оптимумом функции является ее пик (экстремум: максимум или минимум), отклонение от которого влечет нежелательное уменьшение или увеличение значений целевой функции, то есть результата, зависящего от цели оптимизации. Из математического анализа известно, что для целевой функции

$y = f(x)$ от оптимизируемого параметра x оптимумом (экстремумом: максимумом или минимумом) будет точка, удовлетворяющая условию $\partial f(x)/\partial x = 0$. Как правило, все методы оптимизации функции $f(x)$ сводят к методам ее минимизации. Задача максимизации функции $f(x)$ легко преобразуется в минимизацию функции $-f(x)$. Отметим, что функция может (или не может) иметь глобальный (абсолютный) минимум и локальные (местные) минимумы. Большинство методов рассчитано на поиск локальных минимумов. В них область поиска задается определенным диапазоном.

В данном случае *задача оптимизации* сводится к выбору целевой функции по оптимизируемому параметру (или параметрам) и ее обоснованию, записи условий минимума этой целевой функции, составляющих систему уравнений, и решению этой системы уравнений. Данная система уравнений в большинстве практических случаев является нелинейной алгебраической системой, что представляет трудность в ее решении.

Но наибольшую (творческую) трудность составляет выбор целевой функции. Более того, математически сложным для конструктора может оказаться и обоснование этой функции. Вышеизложенный процесс оптимального проектирования, при котором не требуется находить и обосновывать целевую функцию, может быть формализован (записан в виде алгоритма) с привлечением математической теории обобщенных решений. Ясно, что в процессе проектирования необходимо неоднократно решать задачу анализа. Задача анализа была изложена ранее и достаточно формализована. После такой постановки задачи анализа возможно перейти к формализованной постановке задачи оптимального проектирования.

Под *задачей оптимального проектирования* понимается поиск компонентов вектора проектных параметров, удовлетворяющих определенным условиям, зависящих от цели оптимизации. Выбор данных условий, конечно же, представляет не менее трудную творческую задачу, чем выбор целевой функции, однако данные условия хорошо формализуются под конкретный тип задач и в большинстве случаев определены.

Как уже говорилось, после постановки задачи анализа возможна общая постановка задачи оптимального проектирования. Задача оптимального проектирования, как и задача анализа, заключается в решении уравнения деформации конструкции, при этом проектные параметры также являются неизвестными. Согласно математической теории обобщенных решений, необходимо записать условия минимума энергетического функционала – функционала удвоенной полной потенциальной энергии деформации конструкции (функционала Лагранжа), причем не только по перемещениям, но и по проектным параметрам. Условия минимума функционала составят систему нелинейных алгебраических уравнений с единственным тривиальным (нулевым) решением – «лучшая конструкция – это ее отсутствие». Следовательно, необходимо внести **дополнительные условия**, при которых решение будет не тривиальным. Функционал, в обобщенном смысле, выступает в роли целевой функции, при этом оптимизируемые параметры должны удовлетворять дополнительным условиям, что соответствует задаче условной оптимизации. Задачу условной оптимизации сводят к задаче безусловной минимизации с помощью **метода неопределенных множителей Лагранжа**.

Таким образом, задача оптимального проектирования сводится к задаче решения дифференциального уравнения деформации конструкции относительно проектных параметров. С помощью теории обобщенных решений поиск решения уравнения деформации конструкции сводится к условной минимизации соответствующего функционала энергии оператора этого уравнения, при этом математически формулируются дополнительные условия минимизации. С помощью метода неопределенных множителей Лагранжа строится новый функционал, включающий дополнительные условия минимизации, при этом условная оптимизация сводится к безусловной оптимизации нового функционала. Задача безусловной оптимизации функционала сводится к условиям его минимума, которая записывается в виде системы нелинейных алгебраических уравнений, которой трудно рекомендовать какой-либо метод, гарантированно дающий решение. Поэтому вместо решения этой системы применяется **метод прямой минимизации функционала** – метод наис-

корейшего спуска (*градиентный метод*), в котором заложен итерационный процесс, воплощающий итерационный процесс проектирования.

3.2. Метод неопределенных множителей Лагранжа

Как уже говорилось, задача оптимального проектирования заключается в отыскании проектных параметров, доставляющих минимум полной потенциальной энергии деформации конструкции. Минимум (стационарная точка) функционала отыскивается методами *безусловной минимизации*. Однако проектные параметры также должны удовлетворять некоторым дополнительным условиям проектирования. При этом такие условия могут противоречить друг другу. Определение проектных параметров с учетом этих дополнительных условий называется *условной минимизацией* того же энергетического функционала. Чтобы свести задачу условной минимизации функционала к его безусловной минимизации, применяют метод неопределенных множителей Лагранжа.

Суть *метода неопределенных множителей Лагранжа* заключается в том, что дополнительные условия записываются в виде функционалов $\Phi_i(\mathbf{r})=0$, которые равны нулю, при выполнении этих условий. Затем исходный энергетический функционал $F(\mathbf{u}, \mathbf{r})$, модифицируется в следующий функционал:

$$F_1(\mathbf{u}, \mathbf{r}) = F(\mathbf{u}, \mathbf{r}) + \sum_{i=1}^k \lambda_i \Phi_i(\mathbf{r}) + \sum_{j=k+1}^s \int_a^b \lambda_j(x) \Phi_j(x, \mathbf{r}) dx, \quad (3.1)$$

где s – число дополнительных условий проектирования; λ_i – неопределенные множители Лагранжа; $\lambda_j(x)$ – неопределенные функции-множители Лагранжа.

Неопределенные множители Лагранжа λ_i равны нулю, когда не выполняются соответствующие им условия $\Phi_i(\mathbf{r})=0$. Этим они компенсируют невыполнение дополнительных условий. В итоге произведение дополнительных условий на соответствующие неопределенные множители Лагранжа всегда равно нулю и не изменяет значение немодифицированного (исходно-

го) функционала. Значения неопределенных множителей Лагранжа не известны и также подлежат определению.

Далее проводится безусловная минимизация модифицированного функционала (3.1). Следует отметить, что при минимизации функционала (3.1) на сетке Δ , то есть при подстановке в него пробных функций, аппроксимирующих на этой сетке функции перемещений, нагрузки и жесткости (проектной функции), третий член функционала сводится ко второму, при этом

$$\lambda_{i+j} = \int_{x_{i-1}}^{x_i} \lambda_j(x) dx.$$

Условия минимума функционала (3.1) записываются в следующем виде:

$$\frac{\partial F_1(\mathbf{u}, \mathbf{r})}{\partial \mathbf{u}^T} = \frac{\partial F(\mathbf{u}, \mathbf{r})}{\partial \mathbf{u}^T} = 0; \quad \frac{\partial F_1(\mathbf{u}, \mathbf{r})}{\partial \mathbf{r}^T} = 0; \quad \frac{\partial F_1(\mathbf{u}, \mathbf{r})}{\partial \lambda^T} = 0. \quad (3.2)$$

Уравнения (3.2) составляют систему нелинейных алгебраических уравнений, которой трудно рекомендовать какой-либо метод, гарантированно дающий решение. Поэтому функционал (3.1) минимизируют методами его прямой минимизации, например, градиентным методом, который будет рассмотрен позже.

Рассмотрим *дополнительные условия проектирования*, применяемые при получении конструкции максимальной жесткости. В этом случае необходимо записать два дополнительных условия.

В качестве первого дополнительного условия берется *условие перераспределения жесткости*, суть которого заключается в том, что интеграл функции жесткости $C(x)$ по длине должен всегда быть равен заданной величине – исходной суммарной жесткости, то есть заданного среднего значения жесткости C_0 , умноженного на длину конструкции $(b - a)$:

$$\int_a^b C(x) dx = C_0(b - a). \quad (3.3)$$

Условие (3.3) можно записать в функционал, равный нулю:

$$\Phi_0(C) = \int_a^b C(x) dx - C_0(b - a) = 0. \quad (3.4)$$

При минимизации на сетке с кусочно-постоянной функцией жесткости $C(x) = C_i$, $x \in (x_{i-1}, x_i)$ функционал (3.4) записывается в следующем виде:

$$\Phi_0(C) = \sum_{i=1}^n C_i d_i - C_0(b-a) = 0, \quad (3.5)$$

где $d_i = x_i - x_{i-1}$;

а исходный энергетический функционал модифицируется членом

$$\lambda_0 \Phi_0(C) = \lambda_0 \left[\sum_{i=1}^n C_i d_i - C_0(b-a) \right] = 0.$$

В качестве второго дополнительного условия берется **условие равнонапряженности** (равнопрочности), заключающееся в равенстве максимальных напряжений $\sigma(x)$ во всех сечениях x конструкции, при этом все напряжения сравниваются с напряжением в произвольном сечении \bar{x} , принятым в качестве базового. Условие равнонапряженности записывается в виде:

$$|\sigma(x)| = |\sigma(\bar{x})|, \quad \forall (x, \bar{x}) \in (a, b), \quad x \neq \bar{x}. \quad (3.6)$$

Функционал условия (3.2.6) запишется в виде:

$$\Phi_1(\sigma, x) = |\sigma(x)| - |\sigma(\bar{x})| = 0, \quad (3.7)$$

а исходный энергетический функционал модифицируется членом $\int_a^b \lambda_1(x) \Phi_1(\sigma, x) dx = \int_a^b \lambda_1(x) [|\sigma(x)| - |\sigma(\bar{x})|] dx = 0$, который на сетке значений Δ может быть аппроксимирован как:

$$\int_a^b \lambda_1(x) \Phi_1(\sigma, x) dx = \sum_{i=1}^{n-1} \lambda_i [|\sigma(\tilde{x}_i)| - |\sigma(\tilde{x}_{i+1})|] = 0, \quad (3.8)$$

где $\lambda_i = \int_{x_{i-1}}^{x_i} \lambda_1(x) dx$; $\tilde{x}_i = (x_{i-1} + x_i)/2$ – середины участков сетки Δ .

То есть условие (3.6) аппроксимируется условием равенства напряжений в средних сечениях соседних участков и конечных элементов, выполняемого для каждой пары соседних участков (3.8).

Отметим, что напряжения $\sigma(x)$ должны быть выражены через перемещения, например для растяжения сжатия:

$$\sigma(x) = \frac{N(x)}{A(x)} = \frac{EA(x)u'(x)}{A(x)} = E(x)u'(x); \quad (3.9)$$

для кручения:

$$\tau(x) = \frac{T(x)}{I_p(x)} = \frac{GI_p(x)\gamma'(x)}{I_p(x)} = G(x)\gamma'(x); \quad (3.10)$$

для изгиба:

$$\sigma(x) = \frac{M(x)}{I(x)} \xi(x) = \frac{EI(x)v''(x)}{I(x)} \xi(x) = E(x)\xi(x)v''(x). \quad (3.11)$$

3.3. Градиентный метод. Поправки на программном уровне

Условия минимума функционала $F_1(\mathbf{u}, \mathbf{r})$ (3.1) записываются в следующем виде:

$$\frac{\partial F_1(\mathbf{u}, \mathbf{r})}{\partial \mathbf{u}^T} = \frac{\partial F(\mathbf{u}, \mathbf{r})}{\partial \mathbf{u}^T} = 0; \quad \frac{\partial F_1(\mathbf{u}, \mathbf{r})}{\partial \mathbf{r}^T} = 0; \quad \frac{\partial F_1(\mathbf{u}, \mathbf{r})}{\partial \lambda^T} = 0. \quad (3.12)$$

Уравнения (3.12) составляют систему нелинейных алгебраических уравнений, которой трудно рекомендовать какой-либо метод, гарантированно дающий решение. Поэтому функционал (3.1) минимизируют методами его прямой минимизации. В качестве метода прямой минимизации на практике широко используют *градиентный метод*, обладающий простотой. Как известно из курса математического анализа, градиент функции $grad[f(x)] = \partial f / \partial x$ указывает направление возрастания ее значений, следовательно, обратное направление указывает на убывание. Вполне логично в поисках минимума этой функции двигаться в направлении ее убывания, то есть в обратном направлении ее градиента. Однако градиент не указывает на минимум этой функции, и двигаться приходится конечными шагами, поэтому поиск минимума градиентным методом ведется последовательными приближениями.

Итерационная схема градиентного метода для нахождения минимума функционала (3.1) запишется в виде:

$$\mathbf{R}_{(i)} = \mathbf{R}_{(i-1)} - k \frac{\partial F_1(\mathbf{u}, \mathbf{r})}{\partial \mathbf{R}_{(i-1)}^T}, \quad (3.13)$$

где $\mathbf{R} = \{\mathbf{u}, \mathbf{r}, \lambda\}$ – вектор искомых неизвестных; $\partial F_1 / \partial \mathbf{R}$ – градиент модифицированного функционала; k – предохранительный коэффициент; i – порядковый номер итерации.

Компоненты градиента модифицированного функционала составляют левые части однородных уравнений (3.12). Предохранительный коэффициент уменьшает снижение по функционалу в обратном направлении градиенту с тем, чтобы не перескочить минимум, а тем более не удалиться от него. При выборе определенных правил назначения этого коэффициента в зависимости от итераций градиентный метод еще называют методами спуска (например, метод наискорейшего спуска).

То, что компонентами градиента функционала являются выражения из его условий минимума (3.12), также является существенным преимуществом градиентного метода. Следует отметить, что ввиду структуры системы уравнений (3.12) из нее можно выделить задачу анализа, решаемую как систему линейных алгебраических уравнений. В процессе итераций необходимо только уточнять значения векторов проектных параметров и неопределенных множителей Лагранжа, а значения перемещений находить из решения задачи анализа, считая вектор проектных параметров, найденный из предшествующего приближения, заданным.

При реализации итерационного алгоритма такого метода на языке программирования для ЭВМ из дополнительных условий проектирования можно применять поправки на программном уровне, корректирующие значения векторов искомых неизвестных. Тем самым это не позволяет им выходить за область допускаемых значений, что ускоряет процесс отыскания минимума функции и делает алгоритм более надежным.

Выведем *поправку* для изгиба из условия равнонапряженности, используя следующее рассуждение: жесткость последующего приближения (j) будем отыскивать как потребную жесткость для момента предшествующего приближения ($j-1$).

Запишем выражение для напряжений с учетом вышесказанного:

$$\begin{aligned}
 |\sigma(x)| &= \frac{|M^{(j-1)}(x)|}{I^{(j)}(x)} \xi(x) = \frac{EI^{(j-1)}(x)|v''(x)|^{(j-1)}}{I^{(j)}(x)} \xi(x) = \\
 &= \frac{EI^{(j-1)}(x)|v''(x)|^{(j-1)}}{EI^{(j)}(x)} E\xi(x) \leq [\sigma].
 \end{aligned}$$

Условие в правой части показывает, что все напряжения должны не превышать некоторого допускаемого (или разрушающего) напряжения, и в то же время быть близки к нему (быть равными ему при точном решении). Так как напряжения во всех сечениях равны некоторому единственному допускаемому напряжению, то они равны и между собой, следовательно, выполняется условие равнонапряженности. Отсюда находим изгибную жесткость последующего приближения:

$$EI^{(j)}(x) = \frac{EI^{(j-1)}(x)|v''(x)|^{(j-1)}}{[\sigma]} E\xi(x).$$

При аппроксимации функции жесткости на сетке конечных элементов кусочно-постоянной функцией корректировочная формула для значений жесткости запишется в виде:

$$EI_i^{(j)} = \frac{EI_i^{(j-1)}|v_i''^{(j-1)}|}{[\sigma]} E\xi_i, \quad i = 1, 2, \dots, n.$$

Если допускаемое напряжение принять неизвестным, то оно покажет уровень максимальных напряжений, возникающих в конструкции. Найдем его из условия распределения жесткости:

$$\sum_{i=1}^n EI_i^{(j)} d_i = \frac{1}{[\sigma]} \sum_{i=1}^n EI_i^{(j-1)} |v_i''^{(j-1)}| E\xi_i d_i = EI_0(b-a).$$

Отсюда допускаемое напряжение:

$$[\sigma] = \frac{\sum_{i=1}^n EI_i^{(j-1)} |v_i''^{(j-1)}| E\xi_i d_i}{EI_0(b-a)}.$$

Таким образом, в конце каждой итерации градиентного метода вычисляется допускаемое напряжение и подставляется в корректирующую формулу для значений жесткости. В результате значения функции жесткости принудительно корректируются

по двум дополнительным условиям проектирования: условию распределения жесткостей и условию равнонапряженности.

Здесь же принудительно (на программном уровне) можно ограничить значения функции жесткости с тем, чтобы они были больше некоторого заданного значения, выбираемого по конструктивным, технологическим или иным причинам:

$$EI_i \geq EI_{\min} .$$

Отметим, что в некоторой литературе такие поправки на программном уровне иногда называют «кухонными добавками», сравнивая отыскание решения практической задачи проектирования с приготовлением неповторимого изысканного блюда шеф-поваром. Что говорит о сложности и индивидуальности решения нелинейных задач оптимизации, требующих высокой квалификации расчетчика, знаний природы рассматриваемой задачи и методов решения.

3.4. Вопросы для самопроверки

1. *Что такое целевая функция?*
2. *Как записать условия минимума целевой функции?*
3. *Как свести задачу максимизации к задаче минимизации?*
4. *Какие экстремумы может иметь целевая функция?*
5. *К чему сводится задача оптимизации?*
6. *Что такое задача оптимального проектирования?*
7. *Что такое безусловная оптимизация?*
8. *Что такое условная оптимизация?*
9. *Для чего требуются дополнительные условия?*
10. *Каким образом условная оптимизация сводится к безусловной?*
11. *В чем заключается логическая последовательность численных методов оптимального проектирования?*
12. *Что такое итерации?*
13. *В чем заключается итерационный процесс проектирования?*
14. *В чем заключается суть метода неопределенных множителей Лагранжа?*

15. В каком виде записываются дополнительные условия по методу неопределенных множителей Лагранжа?
16. Какие значения принимают неопределенные множители Лагранжа?
17. Что такое модифицированный функционал?
18. Каким методом минимизируется модифицированный функционал?
19. Какими дополнительными условиями проектирования модифицируется функционал?
20. В чем заключается условие перераспределения жесткости?
21. В чем заключается условие равнонапряженности?
22. Как определяются напряжения из перемещений?
23. Что такое методы прямой минимизации функционала?
24. В чем суть градиентного метода?
25. Как записывается итерационная схема градиентного метода?
26. Что включает в себя вектор искомым неизвестных задачи оптимального проектирования?
27. Что включает в себя градиент модифицированного функционала?
28. Для чего используется предохранительный коэффициент в градиентном методе?
29. Каким способом выделяют задачу анализа из задачи оптимального проектирования?
30. Как итерационный алгоритм градиентного метода реализуется на языках программирования для ЭВМ?
31. Что такое поправки на программном уровне?
32. Из каких рассуждений выводится поправка для условия равнонапряженности?
33. Из какого условия и как определяется максимальный уровень напряжений в конструкции?
34. Чему должен быть равен максимальный уровень напряжений в конструкции?

ГЛАВА IV. ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ

4.1. Решение задачи анализа на примере растяжения-сжатия стержня

Перейдем к практическому применению изложенного теоретического материала применительно к задаче анализа. Рассмотрим пример анализа стержня, работающего на растяжение-сжатие, то есть определим его функцию перемещений в зависимости от его геометрических размеров и приложенной к нему нагрузки. Напомним, что дифференциальное уравнение деформации такого стержня записывается следующим образом:

$$Lu = -[EA(x)u'(x)]' = r(x), \quad \forall x \in (a, b), \quad (4.1)$$

где $EA(x)$ – функция жесткости на растяжение-сжатие;

$u(x)$ – искомая функция линейных перемещений поперечных сечений стержня;

$r(x)$ – распределенная (погонная) внешняя растягивающая нагрузка;

ось Ox направлена вдоль оси стержня.

Выберем сетку $\Delta: a = x_0 < x_1 < \dots < x_n = b$, на которой зададим функции жесткости и внешней распределенной нагрузки, и будем на ней искать решение. Зададим (аппроксимируем) функции жесткости и внешней распределенной нагрузки кусочно-постоянными функциями, то есть предположим, что данные функции постоянны на участках (конечных элементах) сетки Δ :

$$EA(x) = EA_i, \quad r(x) = r_i, \quad \forall x \in [x_{i-1}, x_i], \quad i = 1, 2, \dots, n, \quad (4.2)$$

где EA_i и r_i – постоянные значения функций жесткости и внешней распределенной нагрузки на участках (конечных элементах) сетки Δ .

Тогда уравнение (4.1) можно переписать в следующем виде:

$$-EA_i u''(x) = r_i, \quad \forall x \in (x_{i-1}, x_i), \quad i = 1, 2, \dots, n. \quad (4.3)$$

Для получения решения задачи анализа необходимо располагать однородным решением уравнения (4.3), то есть решением

уравнения (4.3) с нулевой правой частью $r_i = 0$ (решением однородного уравнения от (4.3)):

$$-EA_i u''(x) = 0 \Rightarrow u''(x) = 0, \quad \forall x \in (x_{i-1}, x_i), \quad i = 1, 2, \dots, n. \quad (4.4)$$

Решение уравнения (4.4) получается его двукратным неопределенным интегрированием и является многочленом первой степени:

$$u(x) = c_{1,i}x + c_{2,i}, \quad \forall x \in (x_{i-1}, x_i), \quad i = 1, 2, \dots, n, \quad (4.5)$$

где $c_{1,i}$, $c_{2,i}$ – неопределенные константы интегрирования, получающиеся после первого и второго интегрирования, соответственно.

Далее однородное решение (4.5) необходимо разложить на базисные функции, для этого определим константы интегрирования из граничных условий участка (условий интерполирования, условий Эрмита), то есть выразим неизвестные $c_{1,i}$ и $c_{2,i}$ через узловые перемещения u_{i-1} и u_i :

$$c_{1,i}x_{i-1} + c_{2,i} = u_{i-1}; \quad c_{1,i}x_i + c_{2,i} = u_i. \quad (4.6)$$

Таким образом, определив из системы (4.6) неизвестные $c_{1,i}$ и $c_{2,i}$, и подставив их в (4.5), получим однородное решение, разложенное на базисные функции:

$$u(x) = \sum_{j=1}^2 u_{i-2+j} \varphi_{j,i}(x) = u_{i-1} [(x_i - x)/d_i] + u_i [(x - x_{i-1})/d_i], \\ \forall x \in [x_{i-1}, x_i], \quad i = 1, 2, \dots, n, \quad (4.7)$$

где $d_i = x_i - x_{i-1}$ – шаг узлов сетки Δ (длина конечного элемента).

После этого необходимо перейти к функционалу, соответствующему уравнению (4.3), которое является частным случаем уравнения (4.1). Рассмотрим получение функционала в общем виде, то есть функционала, который соответствует общему уравнению (4.1):

$$F(u) = (Lu, u) - 2(r, u) = \left(-[EA(x)u'(x)]', u(x) \right) - 2(r(x), u(x)) = \\ = -\int_a^b [EA(x)u'(x)]' u(x) dx - 2 \int_a^b r(x) u(x) dx. \quad (4.8)$$

В соответствии с сеткой Δ функционал (4.8) может быть разбит на участки (конечные элементы) следующим образом:

$$F(u) = -\sum_{i=1}^n \int_{x_{i-1}}^{x_i} [EA(x)u'(x)] u(x) dx - 2 \sum_{i=1}^n \int_{x_{i-1}}^{x_i} r(x)u(x) dx. \quad (4.9)$$

Для дальнейших преобразований запишем исходные положения – условия непрерывности функции перемещений и внутренних силовых факторов – осевой (растягивающей) силы.

Условие непрерывности функции перемещений $u(x)$ запишется в виде:

$$u(x_i - 0) = u(x_i + 0) = u_i, \quad i = 1, 2, \dots, n-1. \quad (4.10)$$

Условие непрерывности осевой силы $N(x) = -\int r(x) dx = EA(x)u'(x)$ запишется:

$$EA(x_i - 0)u'(x_i - 0) = EA(x_i + 0)u'(x_i + 0), \quad i = 1, 2, \dots, n-1. \quad (4.11)$$

Рассмотрим возможные краевые условия – условия закрепления стержня. Если стержень закреплен в сечении при $x = a$ или $x = b$, тогда:

$$u(x) = 0, \quad x = a, b. \quad (4.12)$$

Если стержень свободен в сечении при $x = a$ или $x = b$, тогда:

$$N(x) = EA(x)u'(x) = 0, \quad x = a, b. \quad (4.13)$$

Далее преобразуем первый член функционала $-\sum_{i=1}^n \int_{x_{i-1}}^{x_i} [EA(x)u'(x)] u(x) dx$, используя формулу интегрирования

по частям $\int_{x_{i-1}}^{x_i} U dV = UV|_{x_{i-1}}^{x_i} - \int_{x_{i-1}}^{x_i} V dU$. Положим $U = u(x)$,

$dV = [EA(x)u'(x)] dx = d[EA(x)u'(x)]$, тогда $V = EA(x)u'(x)$,

$dU = d[u(x)] = u'(x) dx$, и получим:

$$\begin{aligned} & -\sum_{i=1}^n \int_{x_{i-1}}^{x_i} [EA(x)u'(x)] u(x) dx = \\ & = -\sum_{i=1}^n u(x)EA(x)u'(x) \Big|_{x_{i-1}}^{x_i} + \sum_{i=1}^n \int_{x_{i-1}}^{x_i} EA(x)u'(x)u'(x) dx = \\ & = \sum_{i=1}^n \int_{x_{i-1}}^{x_i} EA(x)(u'(x))^2 dx - \end{aligned}$$

$$- \sum_{i=1}^n [EA(x_i)u'(x_i)u(x_i) - EA(x_{i-1})u'(x_{i-1})u(x_{i-1})]. \quad (4.14)$$

Второй член выражения (4.14), используя условие непрерывности функции перемещений (4.10), перепишем в следующем виде:

$$\begin{aligned} & - \sum_{i=1}^n [EA(x_i)u'(x_i)u(x_i) - EA(x_{i-1})u'(x_{i-1})u(x_{i-1})] = \\ & = \sum_{i=1}^{n-1} [EA(x_i + 0)u'(x_i + 0) - EA(x_i - 0)u'(x_i - 0)]u_i - \\ & \quad - [EA(x_n)u'(x_n)u(x_n) - EA(x_0)u'(x_0)u(x_0)]. \end{aligned} \quad (4.15)$$

Введем обозначение узловых реакций:

$$\begin{aligned} R_i & = N(x_i - 0) - N(x_i + 0) = \\ & = EA(x_i - 0)u'(x_i - 0) - EA(x_i + 0)u'(x_i + 0), \quad i = 1, 2, \dots, n-1. \end{aligned} \quad (4.16)$$

Подставим (4.15) с учетом (4.16) в (4.14) и затем в функционал (4.9), получим:

$$\begin{aligned} F(u) & = \sum_{i=1}^n \int_{x_{i-1}}^{x_i} EA(x)(u'(x))^2 dx - 2 \sum_{i=1}^n \int_{x_{i-1}}^{x_i} r(x)u(x)dx - \\ & \quad - \sum_{i=1}^{n-1} R_i u_i - [EA(x_n)u'(x_n)u(x_n) - EA(x_0)u'(x_0)u(x_0)]. \end{aligned} \quad (4.17)$$

Если к стержню не приложены сосредоточенные силы, что соответствует универсальному алгоритму метода конечных элементов, когда сосредоточенные силы распределяются по длине конечных элементов, то справедливо условие непрерывности осевой силы (4.11) и возможны краевые условия только из (4.12) и (4.13). В этом случае реакции из условия непрерывности осевой силы (4.11) равны нулю $R_i = 0$, и для любой комбинации краевых условий из (4.12) и (4.13) функционал (4.17) запишется в виде:

$$F(u) = \sum_{i=1}^n \int_{x_{i-1}}^{x_i} EA(x)(u'(x))^2 dx - 2 \sum_{i=1}^n \int_{x_{i-1}}^{x_i} r(x)u(x)dx. \quad (4.18)$$

Подставляя в функционал (4.18), соответствующий общему уравнению (4.1), аппроксимированные функции (4.2), получим функционал, соответствующий частному уравнению (4.3):

$$F(u) = \sum_{i=1}^n EA_i \int_{x_{i-1}}^{x_i} (u'(x))^2 dx - 2 \sum_{i=1}^n r_i \int_{x_{i-1}}^{x_i} u(x)dx. \quad (4.19)$$

Функционал (4.19) аналогичен удвоенной полной потенциальной энергии деформации стержня кусочно-постоянной жесткости, работающего на растяжение-сжатие под действием

внешней распределенной растягивающей нагрузки кусочно-постоянной интенсивности.

Далее в функционал (4.19) необходимо подставить однородное решение уравнения (4.3), разложенное на базисные функции (4.7). Делать это наиболее удобно в векторно-матричном виде. Представим однородное решение (4.7) и его производную в векторно-матричном виде:

$$\begin{aligned} u(x) &= [u_{i-1} \quad u_i] \cdot \begin{bmatrix} \varphi_{1,i}(x) \\ \varphi_{2,i}(x) \end{bmatrix} = [u_{i-1} \quad u_i] \cdot \begin{bmatrix} (x_i - x)/d_i \\ (x - x_{i-1})/d_i \end{bmatrix}, \\ u'(x) &= [u_{i-1} \quad u_i] \cdot \begin{bmatrix} \varphi'_{1,i}(x) \\ \varphi'_{2,i}(x) \end{bmatrix} = [u_{i-1} \quad u_i] \cdot \begin{bmatrix} -1/d_i \\ 1/d_i \end{bmatrix}, \quad \forall x \in [x_{i-1}, x_i], \\ & \qquad \qquad \qquad i = 1, 2, \dots, n, \end{aligned} \tag{4.20}$$

Подставим (4.20) в (4.19), получим:

$$\begin{aligned} F(u) &= \sum_{i=1}^n EA \int_{x_{i-1}}^{x_i} [u_{i-1} \quad u_i] \cdot \begin{bmatrix} -1/d_i \\ 1/d_i \end{bmatrix} \cdot [-1/d_i \quad 1/d_i] \cdot \begin{bmatrix} u_{i-1} \\ u_i \end{bmatrix} dx - \\ & \quad - 2 \sum_{i=1}^n r_i \int_{x_{i-1}}^{x_i} [u_{i-1} \quad u_i] \cdot \begin{bmatrix} (x_i - x)/d_i \\ (x - x_{i-1})/d_i \end{bmatrix} dx = \\ &= F(u) = \sum_{i=1}^n [u_{i-1} \quad u_i] \cdot \left\{ EA \int_{x_{i-1}}^{x_i} \begin{bmatrix} 1/d_i^2 & -1/d_i^2 \\ -1/d_i^2 & 1/d_i^2 \end{bmatrix} dx \right\} \cdot \begin{bmatrix} u_{i-1} \\ u_i \end{bmatrix} - \\ & \quad - 2 \sum_{i=1}^n [u_{i-1} \quad u_i] \cdot \left\{ r_i \int_{x_{i-1}}^{x_i} \begin{bmatrix} (x_i - x)/d_i \\ (x - x_{i-1})/d_i \end{bmatrix} dx \right\} = \\ &= F(u) = \sum_{i=1}^n [u_{i-1} \quad u_i] \cdot \begin{bmatrix} \frac{EA}{d_i} & -\frac{EA}{d_i} \\ -\frac{EA}{d_i} & \frac{EA}{d_i} \end{bmatrix} \cdot \begin{bmatrix} u_{i-1} \\ u_i \end{bmatrix} - \\ & \quad - 2 \sum_{i=1}^n [u_{i-1} \quad u_i] \cdot \begin{bmatrix} \frac{r_i d_i}{2} \\ \frac{r_i d_i}{2} \end{bmatrix}. \end{aligned} \tag{4.21}$$

В последнем выражении (4.21) видны квадратная матрица – матрица жесткости и вектор-столбец – вектор эквивалентных узловых усилий.

Перемножим вектора и матрицы в выражении (4.21), получим выражение функционала в удобном виде для численных расчетов:

$$F(u) = \sum_{i=1}^n \left\{ \left[\frac{EA_i}{d_i} u_{i-1} - \frac{EA_i}{d_i} u_i \right] u_{i-1} + \left[-\frac{EA_i}{d_i} u_{i-1} + \frac{EA_i}{d_i} u_i \right] u_i \right\} - 2 \sum_{i=1}^n \left\{ \frac{r_i d_i}{2} u_{i-1} + \frac{r_i d_i}{2} u_i \right\}. \quad (4.22)$$

Далее необходимо рассмотреть условия минимума функционала (4.22):

$$\begin{aligned} \frac{\partial F(u)}{\partial u_0} &= \frac{2EA_1}{d_1} u_0 - \frac{2EA_1}{d_1} u_1 - 2 \frac{r_1 d_1}{2} = 0; \\ \frac{\partial F(u)}{\partial u_i} &= -\frac{2EA_i}{d_i} u_{i-1} + 2 \left(\frac{EA_i}{d_i} + \frac{EA_{i+1}}{d_{i+1}} \right) u_i - \frac{2EA_{i+1}}{d_{i+1}} u_{i-1} - \\ &\quad - 2 \left(\frac{r_i d_i}{2} + \frac{r_{i+1} d_{i+1}}{2} \right) = 0, \quad i = 1, 2, \dots, n-1; \\ \frac{\partial F(u)}{\partial u_n} &= -\frac{2EA_n}{d_n} u_{n-1} + \frac{2EA_n}{d_n} u_n - 2 \frac{r_n d_n}{2} = 0. \end{aligned} \quad (4.23)$$

Условия (4.23) образуют систему линейных алгебраических уравнений с трехдиагональной матрицей:

$$\begin{aligned} \frac{EA_1}{d_1} u_0 - \frac{EA_1}{d_1} u_1 &= \frac{r_1 d_1}{2}; \\ -\frac{EA_i}{d_i} u_{i-1} + \left(\frac{EA_i}{d_i} + \frac{EA_{i+1}}{d_{i+1}} \right) u_i - \frac{EA_{i+1}}{d_{i+1}} u_{i-1} &= \frac{r_i d_i}{2} + \frac{r_{i+1} d_{i+1}}{2}, \\ i &= 1, 2, \dots, n-1; \\ -\frac{EA_n}{d_n} u_{n-1} + \frac{EA_n}{d_n} u_n &= \frac{r_n d_n}{2}. \end{aligned} \quad (4.24)$$

В систему (4.24) необходимо подставить краевые условия из (4.12) и (4.13). Достаточно, чтобы стержень был закреплен в одном из его сечений.

Рассмотрим, когда стержень закреплен в левом конце при $x = a$, в этом случае необходимо заменить в системе (4.24) первое уравнение ($\frac{\partial F(u)}{\partial u_0} = 0$) на уравнение $u_0 = 0$ (при закреплении в правом конце при $x = b$ нужно заменить последнее уравнение ($\frac{\partial F(u)}{\partial u_n} = 0$) на уравнение $u_n = 0$).

Для решения системы (4.24) наиболее удобен метод алгебраической прогонки, согласно которому система может быть записана в виде:

$$A_i x_{i-1} - B_i x_i + C_i x_{i+1} = D_i, \quad i = 0, 1, \dots, n,$$

где в соответствии с выражениями (4.24) и краевыми условиями заполним одномерные массивы следующим образом:

$$A_0 = 0; \quad B_0 = -1; \quad C_0 = 0; \quad D_0 = 0$$

(краевое условие – стержень закреплен на левом конце: $u_0 = 0$);

$$A_i = -\frac{EA_i}{d_i}; \quad B_i = -\left(\frac{EA_i}{d_i} + \frac{EA_{i+1}}{d_{i+1}}\right); \quad C_i = -\frac{EA_{i+1}}{d_{i+1}};$$

$$D_i = \frac{r_i d_i}{2} + \frac{r_{i+1} d_{i+1}}{2}, \quad i = 1, 2, \dots, n-1;$$

$$A_n = -\frac{EA_n}{d_n}; \quad B_n = \frac{EA_n}{d_n}; \quad C_n = 0; \quad D_n = \frac{r_n d_n}{2}$$

(последнее уравнение – стержень свободен на правом конце).

Полагаем в развязке $\xi_0 = 0$, $\eta_0 = 0$, $u_{n+1} = 0$.

Прямой ход записывается:

$$\xi_{i+1} = C_i / (B_i - A_i \xi_i), \quad \eta_{i+1} = (A_i \eta_i - D_i) / (B_i - A_i \xi_i), \quad i = 0, 1, \dots, n.$$

Обратный ход (вычисляются искомые неизвестные):

$$u_i = \xi_{i+1} u_{i+1} + \eta_{i+1}, \quad i = n, n-1, \dots, 0.$$

4.2. Решение задачи оптимального проектирования на примере получения стержня максимальной жесткости на растяжение-сжатие

После того, как решена задача анализа, а следовательно, получен функционал в виде (4.22), возможно перейти к общей постановке задачи оптимального проектирования.

Для получения стержня максимальной жесткости необходимо сформулировать два дополнительных условия проектирования. Первым условием является условие перераспределения жесткости, которое записывается в следующем виде:

$$\int_a^b EA(x)dx - EA_0(b-a) = 0. \quad (4.25)$$

С учетом принятой сетки $\Delta: a = x_0 < x_1 < \dots < x_n = b$ и кусочно-постоянной функции жесткости (4.2) условие (4.25) переписывается следующим образом:

$$\begin{aligned} \int_a^b EA(x)dx - EA_0(b-a) &= \sum_{i=1}^n \int_{x_{i-1}}^{x_i} EA(x)dx - EA_0(b-a) = \\ &= \sum_{i=1}^n EA_i d_i - EA_0(b-a) = 0. \end{aligned} \quad (4.26)$$

Вторым условием является условие равнонапряженности, которое может быть аппроксимировано как:

$$|\sigma(\tilde{x}_i)| - |\sigma(\tilde{x}_{i+1})| = 0, \quad \tilde{x}_i = (x_{i-1} + x_i)/2, \quad i = 1, 2, \dots, n-1. \quad (4.27)$$

Напряжения должны быть выражены через перемещения и в случае растяжения-сжатия имеют вид:

$$\sigma(x) = \frac{N(x)}{A(x)} = \frac{EA(x)u'(x)}{A(x)} = E(x)u'(x). \quad (4.28)$$

Подставим в (4.28) производную функции перемещений (4.7):

$$u'(x) = (u_i - u_{i-1})/d_i, \quad \forall x \in [x_{i-1}, x_i], \quad i = 1, 2, \dots, n,$$

получим:

$$\sigma(x) = E_i(u_i - u_{i-1})/d_i, \quad \forall x \in [x_{i-1}, x_i], \quad i = 1, 2, \dots, n. \quad (4.29)$$

Выражения для напряжений (4.29) подставим в условия (4.27), получим окончательные расчетные выражения для условий равнонапряженности:

$$E_i|(u_i - u_{i-1})|/d_i - E_{i+1}|(u_{i+1} - u_i)|/d_{i+1} = 0, \quad i = 1, 2, \dots, n-1. \quad (4.30)$$

Далее методом неопределенных множителей Лагранжа модифицируем функционал (4.22) дополнительными условиями (4.26) и (4.30). Модифицированный функционал запишется в виде:

$$\begin{aligned}
 F_1(u) = & \sum_{i=1}^n \left\{ \left[\frac{EA_i}{d_i} u_{i-1} - \frac{EA_i}{d_i} u_i \right] u_{i-1} + \left[-\frac{EA_i}{d_i} u_{i-1} + \frac{EA_i}{d_i} u_i \right] u_i \right\} - \\
 & - 2 \sum_{i=1}^n \left\{ \frac{r_i d_i}{2} u_{i-1} + \frac{r_i d_i}{2} u_i \right\} + \lambda_0 \left[\sum_{i=1}^n EA_i d_i - EA_0 (b - a) \right] + \\
 & + \sum_{i=1}^{n-1} \lambda_i \left[E_i |(u_i - u_{i-1})| / d_i - E_{i+1} |(u_{i+1} - u_i)| / d_{i+1} \right]. \quad (4.31)
 \end{aligned}$$

После записи модифицированного функционала (4.31) необходимо рассмотреть задачу его минимизации градиентным методом, и, следовательно, получить его частные производные по искомым параметрам.

Запишем частные производные функционала (4.31):

$$\begin{aligned}
 \frac{\partial F_1(u)}{\partial u_j} = & \frac{\partial F(u)}{\partial u_j} + \frac{\sum_{i=1}^{n-1} \lambda_i \left[E_i |(u_i - u_{i-1})| / d_i - E_{i+1} |(u_{i+1} - u_i)| / d_{i+1} \right]}{\partial u_j}, \\
 & j = 0, 1, \dots, n; \\
 \frac{\partial F_1(u)}{\partial EA_i} = & \left(\frac{1}{d_i} u_{i-1} - \frac{1}{d_i} u_i \right) u_{i-1} + \left(-\frac{1}{d_i} u_{i-1} + \frac{1}{d_i} u_i \right) u_i + \lambda_0 d_i, \\
 & i = 1, 2, \dots, n; \\
 \frac{\partial F_1(u)}{\partial \lambda_0} = & \sum_{i=1}^n EA_i d_i - EA_0 (b - a); \\
 \frac{\partial F_1(u)}{\partial \lambda_i} = & E_i |(u_i - u_{i-1})| / d_i - E_{i+1} |(u_{i+1} - u_i)| / d_{i+1}, \\
 & i = 1, 2, \dots, n - 1. \quad (4.32)
 \end{aligned}$$

Итерационная схема градиентного метода для нахождения минимума модифицированного функционала (4.31) запишется в виде:

$$\mathbf{R}^{(i)} = \mathbf{R}^{(i-1)} - k \frac{\partial F_1(\mathbf{u}, \mathbf{r})}{\partial \mathbf{R}^{(i-1)T}}, \quad (4.33)$$

где $\mathbf{R} = \{u_0, u_1, \dots, u_n, EA_1, EA_2, \dots, EA_n, \lambda_0, \lambda_1, \dots, \lambda_{n-1}\}$ – вектор иско-
мых неизвестных;

$\partial F_1 / \partial \mathbf{R}$ – градиент модифицированного функционала
(4.31), компонентами которого являются выражения (4.32);

k – предохранительный коэффициент;

i – порядковый номер итерации.

Рассмотрим получение поправки на программном уровне
из условия равнонапряженности. Запишем выражение для на-
пряжений:

$$\begin{aligned} |\sigma(x)| &= \frac{|N^{(j-1)}(x)|}{A^{(j)}(x)} = \frac{EA^{(j-1)}(x)|u'(x)|^{(j-1)}}{A^{(j)}(x)} = \\ &= \frac{EA^{(j-1)}(x)|u'(x)|^{(j-1)}}{EA^{(j)}(x)} E(x) \leq [\sigma]. \end{aligned}$$

Отсюда находим жесткость последующего приближения:

$$EA^{(j)}(x) = \frac{EA^{(j-1)}(x)|u'(x)|^{(j-1)}}{[\sigma]} E(x).$$

При аппроксимации функции жесткости на сетке конечных эле-
ментов кусочно-постоянной функцией корректировочная фор-
мула для значений жесткости запишется в виде:

$$EA_i^{(j)} = \frac{EA_i^{(j-1)}|u_i^{(j-1)}|}{[\sigma]} E_i = \frac{EA_i^{(j-1)}|(u_i - u_{i-1})|^{(j-1)}}{[\sigma]d_i} E_i,$$

$$i = 1, 2, \dots, n.$$

Найдем допускаемое напряжение из условия распределения
жесткости:

$$\sum_{i=1}^n EA_i^{(j)} d_i = \frac{1}{[\sigma]} \sum_{i=1}^n EA_i^{(j-1)} |(u_i - u_{i-1})|^{(j-1)} E_i = EA_0 (b - a).$$

Отсюда допускаемое напряжение:

$$[\sigma] = \frac{\sum_{i=1}^n EA_i^{(j-1)} |(u_i - u_{i-1})|^{(j-1)} E_i}{EA_0 (b - a)}.$$

Здесь же принудительно (на программном уровне) ограни-
чиваются значения функции жесткости, чтобы они были больше

некоторого заданного значения, выбираемого по конструктивным, технологическим или иным причинам:

$$EA_i \geq EA_{\min}.$$

4.3. Пример моделирования обвода кузова легкового автомобиля «ВАЗ – 2106» кубическим сплайном кусочно-постоянной жесткости

Подготовка исходных данных. Подготовку исходных данных будем вести по чертежу общего вида автомобиля «ВАЗ – 2106» (рис. 1).

С чертежа автомобиля (рис. 1) снимем координаты X и Y 59-ти точек, принадлежащих верхней поверхности автомобиля и плоскости его симметрии. Расчетная сетка Δ (взаимное расположение снимаемых точек) представлена на рис. 2.

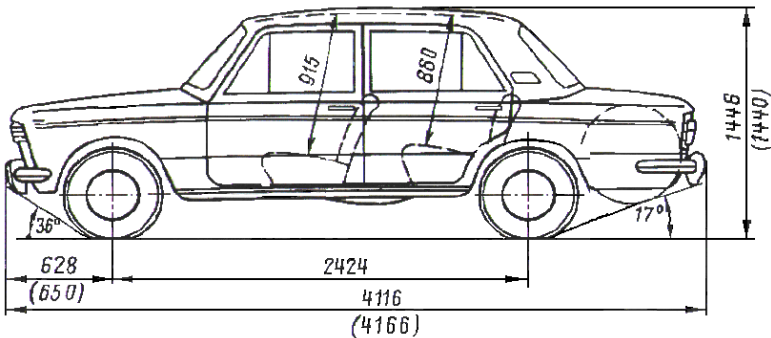


Рис. 1. Чертеж легкового автомобиля «ВАЗ – 2106»

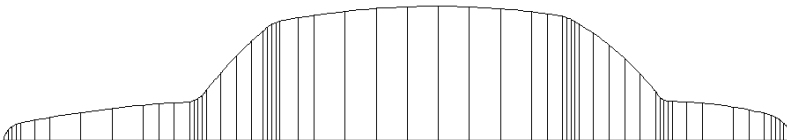


Рис. 2. Расчетная сетка Δ

Координаты снятых точек внесем в табл. 1.

Таблица 1

Исходные данные

Порядковый номер узла i	Координата X_i	Координата Y_i
1	2	3
0	0	0
1	21.664	46.794
2	43.328	67.623
3	64.992	79.693
4	86.656	86.161
5	160.93	100.37
6	241.4	114.73
7	402.33	140.17
8	563.26	161.34
9	724.2	178.26
10	804.66	185.17
11	885.13	191.01
12	965.6	195.81
13	987.26	202
14	1008.9	212.86
15	1030.6	228.34
16	1052.3	256.32
17	1126.5	344.43
18	1207	433.1
19	1287.5	508.55
20	1346.3	560.02
21	1367.9	578
22	1389.6	594.89
23	1411.3	604.33
24	1432.9	609.07
25	1528.9	626.55
26	1609.3	639.31
27	1770.3	662.08
28	1931.2	674.87
29	2092.1	683.53
30	2253.1	686.13
31	2414	682.73
32	2574.9	673.32
33	2735.9	658.74
34	2816.3	649.67
35	2896.8	634.01
36	2918.5	627.24
37	2940.1	618.66

1	2	3
38	2961.8	608.11
39	2983.4	595.48
40	3057.7	547.02
41	3138.2	487.04
42	3218.7	418.05
43	3299.1	337.43
44	3379.6	244.93
45	3401.3	219.02
46	3422.9	204.82
47	3444.6	198.2
48	3466.2	197.33
49	3540.5	194.76
50	3621	189
51	3781.9	168.58
52	3862.4	153.81
53	3942.9	135.93
54	3977.7	125.9
55	3999.4	115.93
56	4021.1	102.59
57	4042.7	84.954
58	4064.4	61.185

На языке программирования М вычислительной среды математической программной системы MATLAB внесение исходных данных записывается следующим образом (имена переменных, могут не совпадать с данными в формулах, что отражено в комментариях):

```
clear;% Очистка памяти
%Исходные данные
X=[0 21.664 43.328 64.992 86.656 160.93 241.4 402.33 563.26 724.2 804.66
885.13 965.6 987.26 1008.9 1030.6 1052.3 1126.5 1207 1287.5 1346.3 1367.9
1389.6 1411.3 1432.9 1528.9 1609.3 1770.3 1931.2 2092.1 2253.1 2414 2574.9
2735.9 2816.3 2896.8 2918.5 2940.1 2961.8 2983.4 3057.7 3138.2 3218.7 3299.1
3379.6 3401.3 3422.9 3444.6 3466.2 3540.5 3621 3781.9 3862.4 3942.9 3977.7
3999.4 4021.1 4042.7 4064.4];%Вектор координат X
Y=[0 46.794 67.623 79.693 86.161 100.37 114.73 140.17 161.34 178.26 185.17
191.01 195.81 202 212.86 228.34 256.32 344.43 433.1 508.55 560.02 578 594.89
604.33 609.07 626.55 639.31 662.08 674.87 683.53 686.13 682.73 673.32 658.74
649.67 634.01 627.24 618.66 608.11 595.48 547.02 487.04 418.05 337.43 244.93
219.02 204.82 198.2 197.33 194.76 189 168.58 153.81 135.93 125.9 115.93 102.59
84.954 61.185];%Вектор координат Y
```


`N=size(X,1);` %Определение количества заданных точек N в единственном столбце 1 вектора X

`plot(X,Y,'ok')`, `axis equal`; %График координат снятых точек для проверки правильности введения в равном масштабе осей (визуализация исходных данных)

Используемые операторы:

; – отмена вывода результата на экран;

% – знак начала комментария, не учитываемого в программе;

= – присваивание;

[]' – задание вектора ([] – задание строки, ' – транспонирование строки в вектор); нумерация элементов вектора начинается с единицы, а не с нуля, как в формулах, поэтому счетчики в программе сдвигаются на единицу ($N = 59$).

Введенные исходные данные показаны графически на рис. 3.

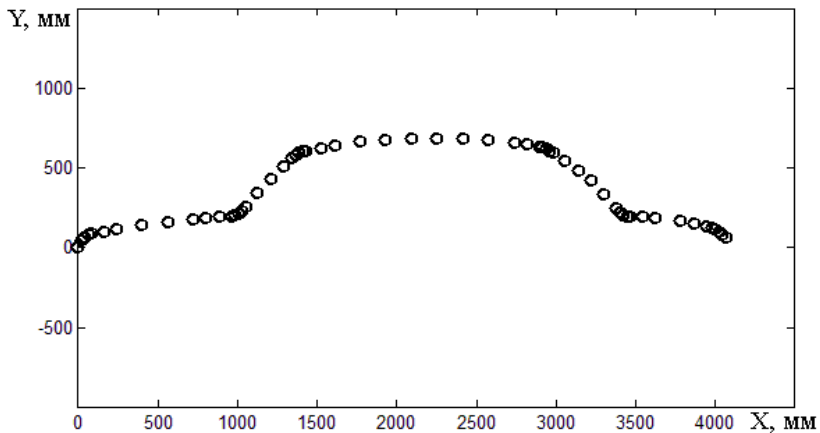


Рис. 3. Визуализация исходных данных

Функция жесткости. Большое число снятых точек позволяет не изменять функцию жесткости сплайна, но для более качественного задания обвода на участках вблизи точек перегиба снизим значения жесткости на 0,1. Занесем значения функции жесткости на участках в табл. 2. Дальнейшее моделирование будем проводить для новых значений.

Таблица 2

Значения функции жесткости на участках

Порядковый номер участка i	Значения функции жесткости на участках при первом моделировании кубическим сплайном $e_i^{(0)}$	Значения функции жесткости на участках после устранения осцилляций e_i
1	2	3
0 [*])	0	0
1	1	0.9
2	1	0.9
3	1	0.9
4	1	0.9
5	1	1
6	1	1
7	1	1
8	1	1
9	1	1
10	1	1
11	1	1
12	1	1
13	1	0.9
14	1	0.9
15	1	0.9
16	1	0.9
17	1	1
18	1	1
19	1	1
20	1	1
21	1	0.9
22	1	0.9
23	1	0.9
24	1	0.9
25	1	1
26	1	1
27	1	1
28	1	1
29	1	1
30	1	1
31	1	1
32	1	1
33	1	1
34	1	1
35	1	1

3. Определим значения параметра x_i в остальных i -х узлах по формуле:

$$x_i = x_{i-1} + \sqrt{(X_i - X_{i-1})^2 + (Y_i - Y_{i-1})^2}; i = 1, 2, \dots, 58.$$

Результат внесем в табл. 3.

4. Тогда правый предел b области определения сетки Δ получается:

$$b = x_{58} = 4478.8.$$

5. Несуществующий шаг узлов d_0 сетки Δ нулевого участка приравняем к нулю:

$$d_0 = 0.$$

6. Шаги узлов d_i сетки Δ для остальных i -х участков вычислим по формуле:

$$d_i = x_i - x_{i-1}, i = 1, 2, \dots, 58.$$

Полученные результаты внесем в табл. 3.

Заметим, что параметр x и шаг узлов d сетки Δ не зависят от функции жесткости e , поэтому при повторном моделировании заполнять табл. 3 не требуется.

Таблица 3

Параметры сетки Δ : $a = 0, b = 4478.8$

Порядковый номер участка i	Параметр x_i	Шаг узлов d_i
1	2	3
0	0	0
1	51.566	51.566
2	81.618	30.053
3	106.42	24.799
4	129.03	22.609
5	204.65	75.621
6	286.39	81.741
7	449.32	162.93
8	611.63	162.32
9	773.46	161.83
10	854.22	80.756
11	934.9	80.682
12	1015.5	80.613
13	1038	22.527

1	2	3
14	1062.3	24.212
15	1088.9	26.656
16	1124.3	35.409
17	1239.5	115.19
18	1359.3	119.76
19	1469.6	110.33
20	1547.7	78.145
21	1575.8	28.104
22	1603.3	27.498
23	1627	23.664
24	1649.1	22.114
25	1746.7	97.578
26	1828.1	81.406
27	1990.7	162.6
28	2152.1	161.41
29	2313.3	161.13
30	2474.3	161.02
31	2635.2	160.94
32	2796.4	161.17
33	2958	161.66
34	3039	80.91
35	3121	82.009
36	3143.7	22.732
37	3166.9	23.242
38	3191.1	24.129
39	3216.1	25.022
40	3304.8	88.707
41	3405.2	100.39
42	3511.2	106.02
43	3625.1	113.86
44	3747.7	122.62
45	3781.5	33.797
46	3807.3	25.85
47	3830	22.687
48	3851.6	21.618
49	3926	74.344
50	4006.7	80.706
51	4168.9	162.19
52	4250.7	81.844
53	4333.2	82.462
54	4369.4	36.217
55	4393.3	23.881

1	2	3
56	4418.7	25.472
57	4446.6	27.885
58	4478.8	32.185

На языке программирования М определение параметров сетки Δ записывается:

```
%Параметры сетки Delta: [a,b]
a=0;%Левый предел области определения
%Параметр t (параметризация по хордам)
t(1)=a;for(I=2:N) t(I)=t(I-1)+sqrt((X(I)-X(I-1))^2+(Y(I)-Y(I-1))^2);end;t=t';
b=t(N);% Правый предел области определения
% Шаг узлов d сетки Delta
d(1)=0;for(I=2:N) d(I)=t(I)-t(I-1);end;d=d';
```

Используемые операторы:

for(...) ... **end** – цикл FOR: действия внутри цикла выполняются, пока счетчик в скобках не пройдет все значения с шагом, по умолчанию равным единице, от начального до конечного указанных там же значений, разделенных двоеточием (при необходимости между начальным и конечным значениями счетчика указывается значение шага, отделенное также двоеточиями);

t(I) – обращение к I-му элементу массива или вектора;

^ – возведение в степень;

sqrt(...) – извлечение квадратного корня.

Определение неизвестных коэффициентов сплайна. Заполним вектор – столбцы системы уравнений для нахождения неизвестных вектор – сплайна методом алгебраической прогонки.

Значения вектор – столбца A вычисляются по формулам:

$$A_0 = 0; A_i = \frac{e_i}{d_i}, i = 1, 2, \dots, n-1; A_n = \frac{1}{d_n}.$$

Значения вектор – столбца B вычисляются по формулам:

$$B_0 = -\frac{e_1 d_2 (d_1 + d_2)}{d_1 (e_2 d_1^2 + 2e_1 d_1 d_2 + e_1 d_2^2)};$$

$$B_i = -2 \left(\frac{e_i}{d_i} + \frac{e_{i+1}}{d_{i+1}} \right), i = 1, 2, \dots, n-1;$$

$$B_n = -\frac{e_n d_{n-1} (d_{n-1} + d_n)}{(e_n d_{n-1}^2 + 2e_n d_{n-1} d_n + e_{n-1} d_n^2) d_n}.$$

Значения вектор – столбца C вычисляются по формулам:

$$C_0 = \frac{1}{d_1}; C_i = \frac{e_{i+1}}{d_{i+1}}, i = 1, 2, \dots, n-1; C_n = 0.$$

Проекция правой части системы на ось X – вектор – столбец DX :

$$\begin{aligned} DX_0 &= 3 \frac{X_1 - X_0}{d_1^2} + \left[\frac{2e_2 d_1 + e_1 d_2}{(e_2 d_1^2 + 2e_1 d_1 d_2 + e_1 d_2^2) d_1} + \frac{1}{d_1^2} \right] X_0 - \\ & - \frac{e_2 d_1^3 + 3e_2 d_1^2 d_2 + 3e_1 d_1 d_2^2 + e_1 d_2^3}{d_1^2 d_2 (e_2 d_1^2 + 2e_1 d_1 d_2 + e_1 d_2^2)} X_1 + \frac{e_2 d_1}{(e_2 d_1^2 + 2e_1 d_1 d_2 + e_1 d_2^2) d_2} X_2; \\ DX_i &= 3e_{i+1} \frac{X_{i+1} - X_i}{d_{i+1}^2} + 3e_i \frac{X_i - X_{i-1}}{d_i^2}, i = 1, 2, \dots, n-1; \\ DX_n &= 3 \frac{X_n - X_{n-1}}{d_n^2} - \frac{e_{n-1} d_n}{(e_n d_{n-1}^2 + 2e_n d_{n-1} d_n + e_{n-1} d_n^2) d_{n-1}} X_{n-2} + \\ & + \frac{e_n d_{n-1}^3 + 3e_n d_{n-1}^2 d_n + 3e_{n-1} d_{n-1} d_n^2 + e_{n-1} d_n^3}{(e_n d_{n-1}^2 + 2e_n d_{n-1} d_n + e_{n-1} d_n^2) d_{n-1} d_n^2} X_{n-1} - \\ & - \left[\frac{e_n d_{n-1} + 2e_{n-1} d_n}{(e_n d_{n-1}^2 + 2e_n d_{n-1} d_n + e_{n-1} d_n^2) d_n} + \frac{1}{d_n^2} \right] X_n. \end{aligned}$$

Проекция правой части системы на ось Y – вектор – столбец DY :

$$\begin{aligned} DY_0 &= 3 \frac{Y_1 - Y_0}{d_1^2} + \left[\frac{2e_2 d_1 + e_1 d_2}{(e_2 d_1^2 + 2e_1 d_1 d_2 + e_1 d_2^2) d_1} + \frac{1}{d_1^2} \right] Y_0 - \\ & - \frac{e_2 d_1^3 + 3e_2 d_1^2 d_2 + 3e_1 d_1 d_2^2 + e_1 d_2^3}{d_1^2 d_2 (e_2 d_1^2 + 2e_1 d_1 d_2 + e_1 d_2^2)} Y_1 + \frac{e_2 d_1}{(e_2 d_1^2 + 2e_1 d_1 d_2 + e_1 d_2^2) d_2} Y_2; \\ DY_i &= 3e_{i+1} \frac{Y_{i+1} - Y_i}{d_{i+1}^2} + 3e_i \frac{Y_i - Y_{i-1}}{d_i^2}, i = 1, 2, \dots, n-1; \\ DY_n &= 3 \frac{Y_n - Y_{n-1}}{d_n^2} - \frac{e_{n-1} d_n}{(e_n d_{n-1}^2 + 2e_n d_{n-1} d_n + e_{n-1} d_n^2) d_{n-1}} Y_{n-2} + \end{aligned}$$

$$\begin{aligned}
& + \frac{e_n d_{n-1}^3 + 3e_n d_{n-1}^2 d_n + 3e_{n-1} d_{n-1} d_n^2 + e_{n-1} d_n^3}{(e_n d_{n-1}^2 + 2e_n d_{n-1} d_n + e_{n-1} d_n^2) d_{n-1} d_n^2} Y_{n-1} - \\
& - \left[\frac{e_n d_{n-1} + 2e_{n-1} d_n}{(e_n d_{n-1}^2 + 2e_n d_{n-1} d_n + e_{n-1} d_n^2) d_n} + \frac{1}{d_n^2} \right] Y_n.
\end{aligned}$$

Полученные результаты внесем в табл. 4.

Таблица 4

Значения вектор – столбцов системы уравнений

Индекс i	Вектор – столбцы (массивы)				
	A_i	B_i	C_i	DX_i	DY_i
1	2	3	4	5	6
0	0	-0.0071407	0.019393	0.013475	0.022419
1	0.017454	-0.094801	0.029947	0.086761	0.10978
2	0.029947	-0.13248	0.036291	0.15987	0.11526
3	0.036291	-0.1522	0.039807	0.20954	0.087153
4	0.039807	-0.10606	0.013224	0.1534	0.041619
5	0.013224	-0.050915	0.012234	0.075095	0.013902
6	0.012234	-0.036743	0.0061377	0.054318	0.0093226
7	0.0061377	-0.024597	0.0061608	0.036512	0.0052856
8	0.0061608	-0.02468	0.0061794	0.036761	0.0043488
9	0.0061794	-0.037125	0.012383	0.055449	0.005117
10	0.012383	-0.049555	0.012394	0.074098	0.0058701
11	0.012394	-0.049599	0.012405	0.074234	0.0049073
12	0.012405	-0.10471	0.039952	0.15239	0.03515
13	0.039952	-0.15425	0.037171	0.21491	0.082952
14	0.037171	-0.14187	0.033764	0.18213	0.10884
15	0.033764	-0.11836	0.025418	0.12919	0.11908
16	0.025418	-0.068198	0.0086812	0.063507	0.080176
17	0.0086812	-0.034062	0.00835	0.033614	0.038468
18	0.00835	-0.034827	0.0090636	0.036677	0.037141
19	0.0090636	-0.043721	0.012797	0.048726	0.04388
20	0.012797	-0.089641	0.032024	0.10272	0.086749
21	0.032024	-0.12951	0.032729	0.15132	0.12177
22	0.032729	-0.14152	0.038032	0.18211	0.10582
23	0.038032	-0.15746	0.040698	0.22388	0.071684
24	0.040698	-0.10189	0.010248	0.1495	0.031678
25	0.010248	-0.045064	0.012284	0.066644	0.011284
26	0.012284	-0.036868	0.00615	0.054665	0.00836
27	0.00615	-0.024691	0.0061955	0.036796	0.0040564
28	0.0061955	-0.024803	0.0062061	0.037119	0.0024734
29	0.0062061	-0.024833	0.0062104	0.03722	0.0013015

Окончание табл. 4

1	2	3	4	5	6
30	0.0062104	-0.024848	0.0062137	0.037265	-0.000093
31	0.0062137	-0.024836	0.0062044	0.037218	-0.0014805
32	0.0062044	-0.024781	0.0061859	0.037064	-0.0027604
33	0.0061859	-0.037091	0.012359	0.055327	-0.0058302
34	0.012359	-0.049106	0.012194	0.072753	-0.011142
35	0.012194	-0.10357	0.039593	0.1493	-0.04236
36	0.039593	-0.15663	0.038724	0.22135	-0.078261
37	0.038724	-0.15205	0.0373	0.2086	-0.091813
38	0.0373	-0.14654	0.035969	0.19379	-0.10339
39	0.035969	-0.094484	0.011273	0.12148	-0.072943
40	0.011273	-0.042469	0.0099613	0.05229	-0.03633
41	0.0099613	-0.038787	0.0094323	0.045449	-0.036269
42	0.0094323	-0.03643	0.0087828	0.040092	-0.037071
43	0.0087828	-0.033876	0.008155	0.034667	-0.037112
44	0.008155	-0.06957	0.02663	0.067356	-0.079702
45	0.02663	-0.12289	0.034817	0.13857	-0.11862
46	0.034817	-0.14897	0.03967	0.20111	-0.092104
47	0.03967	-0.16261	0.041633	0.23863	-0.039753
48	0.041633	-0.11017	0.013451	0.16513	-0.0064215
49	0.013451	-0.051683	0.012391	0.077406	-0.0040479
50	0.012391	-0.037113	0.0061656	0.055427	-0.0049817
51	0.0061656	-0.036768	0.012218	0.054403	-0.0089438
52	0.012218	-0.04869	0.012127	0.071568	-0.014503
53	0.012127	-0.079477	0.027612	0.11511	-0.030829
54	0.027612	-0.1306	0.037687	0.18233	-0.070143
55	0.037687	-0.14604	0.035332	0.19304	-0.10271
56	0.035332	-0.13521	0.032275	0.1653	-0.11675
57	0.032275	-0.12048	0.027964	0.13156	-0.12319
58	0.031071	-0.014423	0	0.031569	-0.032652

На языке программирования М заполнение вектор – столбцов системы уравнений записывается:

% Заполнение матриц системы А, В, С, DX, DY

A(1)=0;

B(1)=-

$E(2)*d(3)*(d(2)+d(3))/(d(2)*(E(3)*(d(2)^2)+2*E(2)*d(2)*d(3)+E(2)*(d(3)^2))$);

C(1)=1/d(2);

DX(1)=3*(X(2)-

$X(1))/(d(2)^2+(E(3)/(E(3)*(d(2)^2)+2*E(2)*d(2)*d(3)+E(2)*(d(3)^2))+E(3)*$

$d(2)+E(2)*d(3))/(E(3)*(d(2)^2)+2*E(2)*d(2)*d(3)+E(2)*(d(3)^2))*d(2))+1/(d(2)^2$

))*X(1)-

$((E(3)*(d(2)^3)+3*E(3)*(d(2)^2)*d(3)+3*E(2)*d(2)*(d(3)^2)+E(2)*(d(3)^3))/(d(2)$

$\wedge^2 * d(3) * (E(3) * (d(2) \wedge^2 + 2 * E(2) * d(2) * d(3) + E(2) * (d(3) \wedge^2))) * X(2) + (E(3) * d(2) / ((E(3) * (d(2) \wedge^2 + 2 * E(2) * d(2) * d(3) + E(2) * (d(3) \wedge^2))) * d(3))) * X(3);$
 $DY(1) = 3 * (Y(2) -$
 $Y(1)) / (d(2) \wedge^2 + (E(3) / (E(3) * (d(2) \wedge^2 + 2 * E(2) * d(2) * d(3) + E(2) * (d(3) \wedge^2))) + (E(3) * d(2) + E(2) * d(3)) / ((E(3) * (d(2) \wedge^2 + 2 * E(2) * d(2) * d(3) + E(2) * (d(3) \wedge^2))) * d(2)) + 1 / (d(2) \wedge^2))) * Y(1) -$
 $((E(3) * (d(2) \wedge^3 + 3 * E(3) * (d(2) \wedge^2) * d(3) + 3 * E(2) * d(2) * (d(3) \wedge^2) + E(2) * (d(3) \wedge^3))) / ((d(2) \wedge^2) * d(3) * (E(3) * (d(2) \wedge^2 + 2 * E(2) * d(2) * d(3) + E(2) * (d(3) \wedge^2)))) * Y(2) + (E(3) * d(2) / ((E(3) * (d(2) \wedge^2 + 2 * E(2) * d(2) * d(3) + E(2) * (d(3) \wedge^2))) * d(3))) * Y(3);$
 $A(N) = 1 / d(N);$
 $B(N) = -E(N) * d(N-1) * (d(N-1) + d(N)) / ((E(N) * (d(N-1) \wedge^2 + 2 * E(N) * d(N-1) * d(N) + E(N-1) * (d(N) \wedge^2)) * d(N));$
 $C(N) = 0;$
 $DX(N) = 3 * (X(N) - X(N-1)) / (d(N) \wedge^2) - (E(N-1) * d(N) / ((E(N) * (d(N-1) \wedge^2 + 2 * E(N) * d(N-1) * d(N) + E(N-1) * (d(N) \wedge^2)) * d(N-1))) * X(N-2) + ((E(N) * (d(N-1) \wedge^3 + 3 * E(N) * (d(N-1) \wedge^2) * d(N) + 3 * E(N-1) * d(N-1) * (d(N) \wedge^2) + E(N-1) * (d(N) \wedge^3)) / ((E(N) * (d(N-1) \wedge^2 + 2 * E(N) * d(N-1) * d(N) + E(N-1) * (d(N) \wedge^2)) * d(N-1) * (d(N) \wedge^2))) * X(N-1) - (E(N-1) / (E(N) * (d(N-1) \wedge^2 + 2 * E(N) * d(N-1) * d(N) + E(N-1) * (d(N) \wedge^2))) + (E(N) * d(N-1) + E(N-1) * d(N)) / ((E(N) * (d(N-1) \wedge^2 + 2 * E(N) * d(N-1) * d(N) + E(N-1) * (d(N) \wedge^2)) * d(N)) + 1 / (d(N) \wedge^2)) * X(N);$
 $DY(N) = 3 * (Y(N) - Y(N-1)) / (d(N) \wedge^2) - (E(N-1) * d(N) / ((E(N) * (d(N-1) \wedge^2 + 2 * E(N) * d(N-1) * d(N) + E(N-1) * (d(N) \wedge^2)) * d(N-1))) * Y(N-2) + ((E(N) * (d(N-1) \wedge^3 + 3 * E(N) * (d(N-1) \wedge^2) * d(N) + 3 * E(N-1) * d(N-1) * (d(N) \wedge^2) + E(N-1) * (d(N) \wedge^3)) / ((E(N) * (d(N-1) \wedge^2 + 2 * E(N) * d(N-1) * d(N) + E(N-1) * (d(N) \wedge^2)) * d(N-1) * (d(N) \wedge^2))) * Y(N-1) - (E(N-1) / (E(N) * (d(N-1) \wedge^2 + 2 * E(N) * d(N-1) * d(N) + E(N-1) * (d(N) \wedge^2))) + (E(N) * d(N-1) + E(N-1) * d(N)) / ((E(N) * (d(N-1) \wedge^2 + 2 * E(N) * d(N-1) * d(N) + E(N-1) * (d(N) \wedge^2)) * d(N)) + 1 / (d(N) \wedge^2)) * Y(N);$
 $for(I=2:(N-1))$
 $A(I) = E(I) / d(I);$
 $B(I) = -2 * (E(I) / d(I) + E(I+1) / d(I+1));$
 $C(I) = E(I+1) / d(I+1);$
 $DX(I) = 3 * E(I+1) * (X(I+1) - X(I)) / (d(I+1) \wedge^2 + 3 * E(I) * (X(I) - X(I-1))) / (d(I) \wedge^2);$
 $DY(I) = 3 * E(I+1) * (Y(I+1) - Y(I)) / (d(I+1) \wedge^2 + 3 * E(I) * (Y(I) - Y(I-1))) / (d(I) \wedge^2);$
 $end;$
 $A=A'; B=B'; C=C'; DX=DX'; DY=DY';$

Решим систему уравнений для нахождения неизвестных вектор – сплайна методом алгебраической прогонки.

Полагаем в развязке:

$$\xi_0 = 0; \eta X_0 = 0; \eta Y_0 = 0; PX_{n+1} = 0; PY_{n+1} = 0.$$

В прямом ходе определим по порядку следующие значения:

$$\xi_{i+1} = \frac{C_i}{(B_i - A_i \xi_i)}; \eta X_{i+1} = \frac{(A_i \eta X_i - DX_i)}{(B_i - A_i \xi_i)};$$

$$\eta Y_{i+1} = \frac{(A_i \eta Y_i - DY_i)}{(B_i - A_i \xi_i)}, i = 0, 1, \dots, n.$$

В обратном ходе вычисляются искомые неизвестные в заданном порядке:

$$PX_i = \xi_{i+1} PX_{i+1} + \eta X_{i+1}; PY_i = \xi_{i+1} PY_{i+1} + \eta Y_{i+1}, i = n, n-1, \dots, 0.$$

Полученные результаты внесем в табл. 5.

Таблица 5

Результат решения системы уравнений

Индекс i	Вспомогательные вектор – столбцы			Искомые неизвестные	
	ξ_i	ηX_i	ηY_i	PX_i	PY_i
1	2	3	4	5	6
0	0	0	0	0.19611	1.0022
1	-2.7158	1.8871	3.1396	0.62265	0.78702
2	-0.63179	1.1355	1.16	0.81178	0.5904
3	-0.31959	1.1084	0.70905	0.92815	0.37126
4	-0.28313	1.2042	0.43686	0.97512	0.2317
5	-0.1395	1.1125	0.2556	0.98498	0.1713
6	-0.24931	1.2305	0.21442	0.98501	0.17297
7	-0.18217	1.1653	0.19884	0.98991	0.14201
8	-0.2624	1.2505	0.17314	0.99295	0.11865
9	-0.26793	1.2599	0.14231	0.99622	0.088296
10	-0.34912	1.3438	0.11947	0.99565	0.089301
11	-0.27402	1.2703	0.097072	1.0023	0.028357
12	-0.26849	1.2659	0.080174	0.98192	0.19299
13	-0.39407	1.3482	0.33689	0.92954	0.36517
14	-0.26838	1.1628	0.50174	0.86898	0.50886
15	-0.25599	1.0532	0.68382	0.71948	0.68347
16	-0.23166	0.85338	0.87487	0.57801	0.82625
17	-0.13932	0.67111	0.92986	0.66817	0.74366
18	-0.25416	0.84583	0.9252	0.699	0.71424
19	-0.27713	0.9055	0.89943	0.74514	0.66824

1	2	3	4	5	6
20	-0.31053	0.98325	0.867	0.76679	0.64005
21	-0.37382	1.0522	0.88311	0.76359	0.65022
22	-0.27846	1.0008	0.79543	0.85172	0.52146
23	-0.28723	1.128	0.6026	0.96178	0.28247
24	-0.27774	1.2351	0.33279	0.98398	0.18121
25	-0.11313	1.0955	0.20017	0.98561	0.16763
26	-0.27979	1.2622	0.21028	0.98859	0.15246
27	-0.18396	1.1714	0.1728	0.99352	0.11058
28	-0.26297	1.2561	0.12707	0.99838	0.062708
29	-0.2678	1.266	0.072761	0.99919	0.037541
30	-0.26803	1.2673	0.03668	1.0001	-0.0032125
31	-0.26802	1.2679	-0.013836	0.99924	-0.039638
32	-0.26777	1.2662	-0.060186	0.99713	-0.076738
33	-0.26756	1.2633	-0.10325	0.9949	-0.099079
34	-0.34879	1.3408	-0.14651	0.9917	-0.13598
35	-0.27221	1.2542	-0.2083	0.96421	-0.26571
36	-0.39492	1.3366	-0.3972	0.94304	-0.33295
37	-0.27464	1.1946	-0.44352	0.9159	-0.40261
38	-0.26377	1.148	-0.52781	0.88001	-0.47467
39	-0.26312	1.1044	-0.61235	0.8527	-0.52325
40	-0.13259	0.9616	-0.59889	0.8213	-0.57047
41	-0.24311	1.0116	-0.7219	0.78283	-0.62285
42	-0.25938	0.97269	-0.7996	0.73199	-0.68143
43	-0.25844	0.90976	-0.8689	0.68784	-0.72536
44	-0.25802	0.84403	-0.93275	0.60534	-0.80374
45	-0.39472	0.89635	-1.0686	0.73727	-0.67107
46	-0.30981	1.0207	-0.80233	0.91475	-0.42369
47	-0.28707	1.1982	-0.46437	0.98731	-0.1417
48	-0.27532	1.2637	-0.14106	1.004	0.0023256
49	-0.13627	1.1399	-0.0055582	0.99751	-0.057853
50	-0.24856	1.2452	-0.079702	0.99647	-0.087905
51	-0.18117	1.1753	-0.11736	0.98698	-0.1626
52	-0.34272	1.3227	-0.23057	0.97965	-0.19833
53	-0.2725	1.245	-0.26259	0.97384	-0.23583
54	-0.36249	1.313	-0.36292	0.93556	-0.3506
55	-0.31253	1.2114	-0.49857	0.88256	-0.47348
56	-0.26316	1.0977	-0.62508	0.81766	-0.57608
57	-0.25632	1.0047	-0.75179	0.7299	-0.68551
$n = 58$	-0.24922	0.88352	-0.88167	0.61637	-0.78712
$n + 1 = 59$	0	0.61637	-0.78712	0	0

На языке программирования М решение системы уравнений методом алгебраической прогонки записывается:

```

% Развязка
KSI(1)=0;ITAX(1)=0;ITAY(1)=0;PX(N+1)=0;PY(N+1)=0;
% Прямой ход
for(I=1:N)
    KSI(I+1)=C(I)/(B(I)-A(I)*KSI(I));
    ITAX(I+1)=(A(I)*ITAX(I)-DX(I))/(B(I)-A(I)*KSI(I));
    ITAY(I+1)=(A(I)*ITAY(I)-DY(I))/(B(I)-A(I)*KSI(I));
end;
% Обратный ход (вычисляются искомые неизвестные PX, PY)
for(I=N:-1:1)
    PX(I)=KSI(I+1)*PX(I+1)+ITAX(I+1);
    PY(I)=KSI(I+1)*PY(I+1)+ITAY(I+1);
end;
KSI=KSI';ITAX=ITAX';ITAY=ITAY';PX=PX';PY=PY';

```

Моделирование обводов легкового автомобиля сплайном.

Заполним табл. 6, взяв ранее вычисленные значения из табл. 1, 2, 3 и 5.

Таблица 6

Расчетные данные для моделирования обвода

Порядковый номер узла i	Координата X_i	Координата Y_i	Параметр x_i	Шаг узлов d_i	Функция жесткости e_i	Проекция производных сплайна	
						PX_i	PY_i
1	2	3	4	5	6	7	8
0	0	0	0	0	0	0.19611	1.0022
1	21.664	46.794	51.566	51.566	0.9	0.62265	0.78702
2	43.328	67.623	81.618	30.053	0.9	0.81178	0.5904
3	64.992	79.693	106.42	24.799	0.9	0.92815	0.37126
4	86.656	86.161	129.03	22.609	0.9	0.97512	0.2317
5	160.93	100.37	204.65	75.621	1	0.98498	0.1713
6	241.4	114.73	286.39	81.741	1	0.98501	0.17297
7	402.33	140.17	449.32	162.93	1	0.98991	0.14201
8	563.26	161.34	611.63	162.32	1	0.99295	0.11865
9	724.2	178.26	773.46	161.83	1	0.99622	0.088296
10	804.66	185.17	854.22	80.756	1	0.99565	0.089301

Продолжение табл. 6

1	2	3	4	5	6	7	8
11	885.13	191.01	934.9	80.682	1	1.0023	0.028357
12	965.6	195.81	1015.5	80.613	1	0.98192	0.19299
13	987.26	202	1038	22.527	0.9	0.92954	0.36517
14	1008.9	212.86	1062.3	24.212	0.9	0.86898	0.50886
15	1030.6	228.34	1088.9	26.656	0.9	0.71948	0.68347
16	1052.3	256.32	1124.3	35.409	0.9	0.57801	0.82625
17	1126.5	344.43	1239.5	115.19	1	0.66817	0.74366
18	1207	433.1	1359.3	119.76	1	0.699	0.71424
19	1287.5	508.55	1469.6	110.33	1	0.74514	0.66824
20	1346.3	560.02	1547.7	78.145	1	0.76679	0.64005
21	1367.9	578	1575.8	28.104	0.9	0.76359	0.65022
22	1389.6	594.89	1603.3	27.498	0.9	0.85172	0.52146
23	1411.3	604.33	1627	23.664	0.9	0.96178	0.28247
24	1432.9	609.07	1649.1	22.114	0.9	0.98398	0.18121
25	1528.9	626.55	1746.7	97.578	1	0.98561	0.16763
26	1609.3	639.31	1828.1	81.406	1	0.98859	0.15246
27	1770.3	662.08	1990.7	162.6	1	0.99352	0.11058
28	1931.2	674.87	2152.1	161.41	1	0.99838	0.062708
29	2092.1	683.53	2313.3	161.13	1	0.99919	0.037541
30	2253.1	686.13	2474.3	161.02	1	1.0001	-0.0032125
31	2414	682.73	2635.2	160.94	1	0.99924	-0.039638
32	2574.9	673.32	2796.4	161.17	1	0.99713	-0.076738
33	2735.9	658.74	2958	161.66	1	0.9949	-0.099079
34	2816.3	649.67	3039	80.91	1	0.9917	-0.13598
35	2896.8	634.01	3121	82.009	1	0.96421	-0.26571
36	2918.5	627.24	3143.7	22.732	0.9	0.94304	-0.33295
37	2940.1	618.66	3166.9	23.242	0.9	0.9159	-0.40261
38	2961.8	608.11	3191.1	24.129	0.9	0.88001	-0.47467
39	2983.4	595.48	3216.1	25.022	0.9	0.8527	-0.52325
40	3057.7	547.02	3304.8	88.707	1	0.8213	-0.57047
41	3138.2	487.04	3405.2	100.39	1	0.78283	-0.62285
42	3218.7	418.05	3511.2	106.02	1	0.73199	-0.68143
43	3299.1	337.43	3625.1	113.86	1	0.68784	-0.72536
44	3379.6	244.93	3747.7	122.62	1	0.60534	-0.80374
45	3401.3	219.02	3781.5	33.797	0.9	0.73727	-0.67107
46	3422.9	204.82	3807.3	25.85	0.9	0.91475	-0.42369
47	3444.6	198.2	3830	22.687	0.9	0.98731	-0.1417
48	3466.2	197.33	3851.6	21.618	0.9	1.004	0.0023256
49	3540.5	194.76	3926	74.344	1	0.99751	-0.057853
50	3621	189	4006.7	80.706	1	0.99647	-0.087905
51	3781.9	168.58	4168.9	162.19	1	0.98698	-0.1626
52	3862.4	153.81	4250.7	81.844	1	0.97965	-0.19833

1	2	3	4	5	6	7	8
53	3942.9	135.93	4333.2	82.462	1	0.97384	-0.23583
54	3977.7	125.9	4369.4	36.217	1	0.93556	-0.3506
55	3999.4	115.93	4393.3	23.881	0.9	0.88256	-0.47348
56	4021.1	102.59	4418.7	25.472	0.9	0.81766	-0.57608
57	4042.7	84.954	4446.6	27.885	0.9	0.7299	-0.68551
58	4064.4	61.185	4478.8	32.185	0.9	0.61637	-0.78712

Построим график обвода легкового автомобиля ВАЗ – 2106 ($SX(x), SY(x)$) по участкам между заданными точками на ЭВМ в программе MATLAB. Координаты точек графика определяются нижеследующими выражениями. Координата X (абсцисса) графика обвода определяется выражением сплайна $SX(x)$ для участка $x_{j-1} \leq x \leq x_j$:

$$SX(x) = X_{j-1} \frac{(x_j - x)^2 [2(x - x_{j-1}) + d_j]}{d_j^3} + X_j \frac{(x - x_{j-1})^2 [2(x_j - x) + d_j]}{d_j^3} + \\ + PX_{j-1} \frac{(x_j - x)^2 (x - x_{j-1})}{d_j^2} - PX_j \frac{(x - x_{j-1})^2 (x_j - x)}{d_j^2};$$

а координата Y (ордината) графика обвода определяется выражением сплайна $SY(x)$ для участка $x_{j-1} \leq x \leq x_j$:

$$SY(x) = Y_{j-1} \frac{(x_j - x)^2 [2(x - x_{j-1}) + d_j]}{d_j^3} + Y_j \frac{(x - x_{j-1})^2 [2(x_j - x) + d_j]}{d_j^3} + \\ + PY_{j-1} \frac{(x_j - x)^2 (x - x_{j-1})}{d_j^2} - PY_j \frac{(x - x_{j-1})^2 (x_j - x)}{d_j^2},$$

где $j = 1, 2, \dots, n$.

На языке программирования М построение графика обвода записывается:

```
% Вычисление значений интерполяционной функции
J=0;% Порядковый номер значения интерполяционной функции
h=1;% Шаг плотности вычисления промежуточных точек h
for(I=2:N)
    for(v=t(I-1):h:t(I))
        J=J+1;
```

```

SX=X(I-1)*((t(I)-v)^2)*(2*(v-t(I-1))+d(I))/(d(I)^3)+X(I)*((v-t(I-1))^2*(2*(t(I)-v)+d(I))/(d(I)^3)+PX(I-1)*((t(I)-v)^2*(v-t(I-1)))/(d(I)^2)-PX(I)*((v-t(I-1))^2*(t(I)-v))/(d(I)^2));
SY=Y(I-1)*((t(I)-v)^2)*(2*(v-t(I-1))+d(I))/(d(I)^3)+Y(I)*((v-t(I-1))^2*(2*(t(I)-v)+d(I))/(d(I)^3)+PY(I-1)*((t(I)-v)^2*(v-t(I-1)))/(d(I)^2)-PY(I)*((v-t(I-1))^2*(t(I)-v))/(d(I)^2));
x(J)=SX;
y(J)=SY;
end;
end;
x=x';y=y';
% График сплайна
plot(x,y,X,Y,'ok'),axis equal;

```

Полученная модель обвода представлена на рис. 4 и 5.

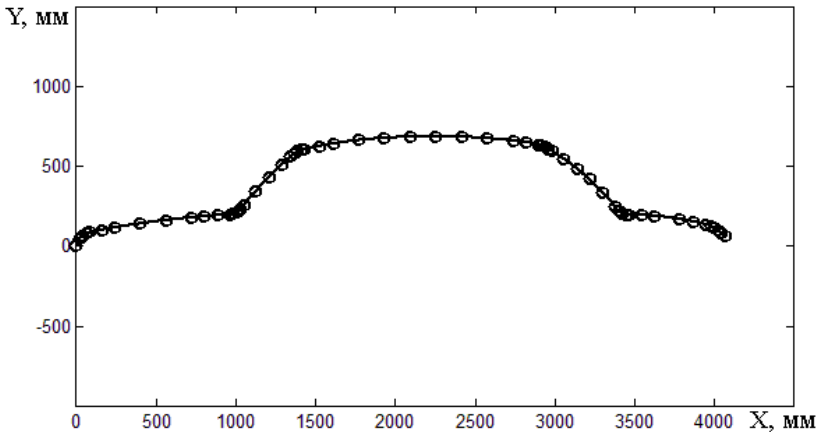


Рис. 4. Моделируемый обвод с исходными данными

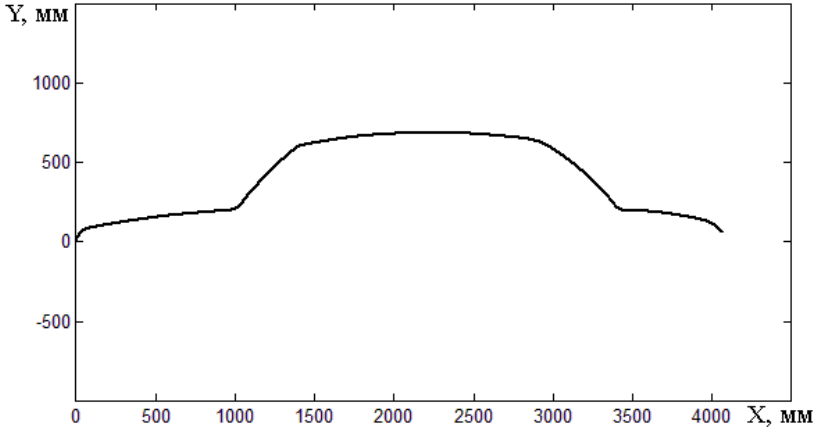


Рис. 5. Моделируемый обвод

4.4. Пример анализа балки передней оси грузового автомобиля «КАМАЗ – 65115» методом конечных элементов на изгиб

Подготовка исходных данных. Подготовку исходных данных будем вести по чертежу балки передней оси грузового автомобиля «КАМАЗ – 65115». Балка шарнирно оперта на колеса автомобиля и нагружена двумя равными поперечными силами на поддрессорных площадках от веса автомобиля, приходящегося на его переднюю ось. На рис. 6 представлены схематическое изображение балки, разбитой на конечные элементы (КЭ), расчетная схема балки, внешние изгибающие нагрузки и эпюра жесткости балки, снятая с ее чертежа.

Для остальных конечных элементов средние их жесткости совпадают с узловыми значениями эпюры изгибной жесткости балки и дополнительного определения не требуют.

Внешняя нагрузка приложена только к 4-му и 7-му КЭ в виде сосредоточенных сил, которые распределим по длине конечных элементов следующим образом (с учетом знака):

$$q_4 = \frac{F}{x_4 - x_3} = \frac{-2,5 * 10^3}{0,5 - 0,425} = -33333,3333;$$

$$q_7 = \frac{F}{x_7 - x_6} = \frac{-2,5 * 10^3}{1,175 - 1,1} = -33333,3333.$$

На остальных конечных элементах распределенная внешняя нагрузка отсутствует, т.е. равна нулю.

Таблица 7

Исходные данные для анализа балки МКЭ

Порядковый номер КЭ i	Координата начала КЭ x_{i-1}	Координата конца КЭ x_i	Изгибная жесткость КЭ e_i	Распределенная внешняя нагрузка на КЭ q_i
1	0	0.05	5000	0
2	0.05	0.4	11000	0
3	0.4	0.425	17000	0
4	0.425	0.5	20000	-33333.3333
5	0.5	0.8	17000	0
6	0.8	1.1	17000	0
7	1.1	1.175	20000	-33333.3333
8	1.175	1.2	17000	0
9	1.2	1.55	11000	0
10	1.55	1.6	5000	0

Таким образом, получены все необходимые данные для анализа балки на изгиб методом конечных элементов.

На языке программирования М вычислительной среды математической программной системы MATLAB внесение исходных данных записывается следующим образом (имена переменных могут не совпадать с данными в формулах, что отражено в комментариях):

```
clear% Очистка памяти
% Исходные данные
```

$X=[0 \ 0.05 \ 0.4 \ 0.425 \ 0.5 \ 0.8 \ 1.1 \ 1.175 \ 1.2 \ 1.55 \ 1.6]'$; %X - координаты расчетных сечений по длине балки
 $E=[0 \ 5000 \ 11000 \ 17000 \ 20000 \ 17000 \ 20000 \ 17000 \ 11000 \ 5000]'$; %E - значения изгибной жесткости конечных элементов
 $Q=[0 \ 0 \ 0 \ 0 \ -33333.3333 \ 0 \ 0 \ -33333.3333 \ 0 \ 0 \ 0]'$; %Q - погонная нагрузка
 $N=size(X,1)$; %Определение количества заданных точек N в единственном столбце 1 вектора X

Система уравнений МКЭ для анализа балочной конструкции. Краевыми условиями балки передней оси автомобиля «КамАЗ – 65115» являются шарнирные опоры на обоих концах. Составим для нее нижеследующую систему уравнений.

Краевые условия для левого конца балки (шарнирная опора):

$$V_0 = 0;$$

$$e_1 \left(\frac{12}{d_1^2} V_0 + \frac{8}{d_1} V_0' + \frac{-12}{d_1} V_1 + \frac{4}{d_1} V_1' \right) = \frac{q_1 d_1^2}{6}.$$

Уравнения для внутренних конечных элементов:

$$\begin{aligned} & \frac{-24e_j}{d_j^3} V_{j-1} + \frac{-12e_j}{d_j^2} V_{j-1}' + 24 \left(\frac{e_j}{d_j^3} + \frac{e_{j+1}}{d_{j+1}^3} \right) V_j + 12 \left(\frac{-e_j}{d_j^2} + \frac{e_{j+1}}{d_{j+1}^2} \right) V_j' + \\ & + \frac{-24e_{j+1}}{d_{j+1}^3} V_{j+1} + \frac{12e_{j+1}}{d_{j+1}^2} V_{j+1}' = q_j d_j + q_{j+1} d_{j+1}; \\ & \frac{12e_j}{d_j^2} V_{j-1} + \frac{4e_j}{d_j} V_{j-1}' + 12 \left(\frac{-e_j}{d_j^2} + \frac{e_{j+1}}{d_{j+1}^2} \right) V_j + 8 \left(\frac{e_j}{d_j} + \frac{e_{j+1}}{d_{j+1}} \right) V_j' + \\ & + \frac{-12e_{j+1}}{d_{j+1}^2} V_{j+1} + \frac{4e_{j+1}}{d_{j+1}} V_{j+1}' = -\frac{q_j d_j^2}{6} + \frac{q_{j+1} d_{j+1}^2}{6}, j = 1, 2, \dots, n-1. \end{aligned}$$

Краевые условия для правого конца балки (шарнирная опора):

$$V_n = 0;$$

$$e_n \left(\frac{12}{d_n^2} V_{n-1} + \frac{4}{d_n} V_{n-1}' + \frac{-12}{d_n^2} V_n + \frac{8}{d_n} V_n' \right) = -\frac{q_n d_n^2}{6}.$$

Так же следует вычислить длины конечных элементов d_i по формуле:

$$d_0 = 0; \quad d_i = x_i - x_{i-1}, i = 1, 2, \dots, 58.$$

Полученные значения длин конечных элементов d_i следует внести в табл. 10.

Заполним матрицы системы. Результаты вычислений внесем в табл. 8.

Таблица 8

Система уравнений МКЭ (фрагмент)

Индексы элементов i/j	Матрица \mathbf{A}				Вектор \mathbf{R}	Вектор \mathbf{B}
	0	1	2	3		
0	1	0	0	0	V_0	0
1	2.4e+007	8e+005	-2.4e+007	4e+005	V_0^*	0
2	-9.6e+008	-2.4e+007	9.6616e+008	-2.2922e+007	V_1	0
3	2.4e+007	4e+005	-2.2922e+007	1.0514e+006	V_1^*	0
4	0	0	-6.1574e+006	-1.0776e+006	V_2	0
5	0	0	1.0776e+006	1.2571e+005	V_2^*	0
6	0	0	0	0	V_3	-2500
7	0	0	0	0	V_3^*	-31.25
8	0	0	0	0	V_4	-2500
9	0	0	0	0	V_4^*	31.25
10	0	0	0	0	V_5	0
11	0	0	0	0	V_5^*	0
12	0	0	0	0	V_6	-2500
13	0	0	0	0	V_6^*	-31.25
14	0	0	0	0	V_7	-2500
15	0	0	0	0	V_7^*	31.25
16	0	0	0	0	V_8	0
17	0	0	0	0	V_8^*	0
18	0	0	0	0	V_9	0
19	0	0	0	0	V_9^*	0
2n=20	0	0	0	0	V_{10}	0
2n+1=21	0	0	0	0	V_{10}^*	0

На языке программирования М заполнение матриц системы уравнений записывается:

```

% Шаг узлов сетки Delta D
D(1)=0;for(I=2:N)D(I)=X(I)-X(I-1);end;D=D';
% Обнуление матриц A и B
A=zeros(2*N);B=zeros(2*N,1);
% Заполнение матриц A и B
J=3;
for(I=2:N-1)
    A(J,J-2)=-24*E(I)/(D(I)^3);
    A(J,J-1)=-12*E(I)/(D(I)^2);
    A(J,J)=24*(E(I)/(D(I)^3)+E(I+1)/(D(I+1)^3));
    A(J,J+1)=12*(-E(I)/(D(I)^2)+E(I+1)/(D(I+1)^2));
    A(J,J+2)=-24*E(I+1)/(D(I+1)^3);
    A(J,J+3)=12*E(I+1)/(D(I+1)^2);
    B(J)=Q(I)*D(I)+Q(I+1)*D(I+1);
    A(J+1,J-2)=12*E(I)/(D(I)^2);
    A(J+1,J-1)=4*E(I)/D(I);
    A(J+1,J)=12*(-E(I)/(D(I)^2)+E(I+1)/(D(I+1)^2));
    A(J+1,J+1)=8*(E(I)/D(I)+E(I+1)/D(I+1));
    A(J+1,J+2)=-12*E(I+1)/(D(I+1)^2);
    A(J+1,J+3)=4*E(I+1)/D(I+1);
    B(J+1)=-Q(I)*(D(I)^2)/6+Q(I+1)*(D(I+1)^2)/6;
    J=J+2;
end;
% Краевые условия (балка закреплена с обоих концов)
A(1,1)=1;
A(2,1)=12*E(2)/(D(2)^2);
A(2,2)=8*E(2)/D(2);
A(2,3)=-12*E(2)/(D(2)^2);
A(2,4)=4*E(2)/D(2);
B(2)=Q(2)*(D(2)^2)/6;
A(2*N-1,2*N-1)=1;
A(2*N,2*N-3)=12*E(N)/(D(N)^2);
A(2*N,2*N-2)=4*E(N)/D(N);
A(2*N,2*N-1)=-12*E(N)/(D(N)^2);
A(2*N,2*N)=8*E(N)/D(N);
B(2*N)=-Q(N)*(D(N)^2)/6;

```

Решим систему уравнений методом Гаусса для разреженных матриц с выбором ведущих элементов по столбцам в программе MATLAB (теоретическое решение системы: $\mathbf{R} = \mathbf{A} \setminus \mathbf{B}$)

и распределим вектор искомым неизвестных \mathbf{R} по неизвестным V_i, V_i' . Результат внесем в табл. 9.

Таблица 9

Обобщенные перемещения балки

Номер узла i	Прогиб V_i	Производная прогиба V_i'
0	0	-0.044662
1	-0.0022227	-0.044037
2	-0.015316	-0.02614
3	-0.01595	-0.024623
4	-0.017641	-0.020404
5	-0.020702	0
6	-0.017641	0.020404
7	-0.01595	0.024623
8	-0.015316	0.02614
9	-0.0022227	0.044037
10	0	0.044662

Симметрия балки и нагрузок на нее может служить проверкой правильности решения, в этом случае прогибы и их производные также должны быть симметричны, а также производная прогиба в центральном узле должна равняться нулю, что присутствует в решении.

На языке программирования М решение системы уравнений методом Гаусса для разреженных матриц с выбором ведущих элементов по столбцам и распределение неизвестных записывается:

```
% Решение системы уравнений методом Гаусса для разреженной матрицы
A=sparse(A);% Разрежение матрицы A для ускорения вычислений
R=A\B;
% Распределение
J=1;
for(I=1:N)
    Y(I)=R(I+J-1);% Прогиб
    PY(I)=R(I+J);% Производная прогиба
    J=J+1;
end;
```

Построение функции прогиба балочной конструкции.

Сведем в табл. 10 все необходимые данные для моделирования прогиба балки, которые вычислены ранее.

Таблица 10

Расчетные данные прогиба балки

Но- мер узла i	Пара- метр x_i	Изгиб- ная жест- кость КЭ e_i	Распреде- ленная внешняя нагрузка на КЭ q_i	Шаг узлов $d_i = x_i - x_{i-1}$	Прогиб V_i	Производ- ная проги- ба V'_i
0	0	0	0	0	0	-0.044662
1	0.05	5000	0	0.05	-0.0022227	-0.044037
2	0.4	11000	0	0.35	-0.015316	-0.02614
3	0.425	17000	0	0.025	-0.01595	-0.024623
4	0.5	20000	-33333.3333	0.075	-0.017641	-0.020404
5	0.8	17000	0	0.3	-0.020702	0
6	1.1	17000	0	0.3	-0.017641	0.020404
7	1.175	20000	-33333.3333	0.075	-0.01595	0.024623
8	1.2	17000	0	0.025	-0.015316	0.02614
9	1.55	11000	0	0.35	-0.0022227	0.044037
10	1.6	5000	0	0.05	0	0.044662

Теперь по имеющимся расчетным данным (табл. 10) построим на ЭВМ в программе MATLAB график прогиба балки $(x, V(x))$ по участкам между узлами, разбивающими балку на конечные элементы. За параметр взята координата x (абсцисса) графика прогиба, а координата y (ордината) графика определяется выражением функции прогиба $V(x)$ для участка $x_{j-1} \leq x \leq x_j$:

$$V(x) = \frac{(x_j - x)^2 [2(x - x_{j-1}) + d_j]}{d_j^3} V_{j-1} + \frac{(x - x_{j-1})^2 [2(x_j - x) + d_j]}{d_j^3} V_j + \\ + \frac{(x_j - x)^2 (x - x_{j-1})}{d_j^2} V'_{j-1} - \frac{(x - x_{j-1})^2 (x_j - x)}{d_j^2} V'_j,$$

где $j = 1, 2, \dots, n$.

На языке программирования М построение графика функции прогиба балки записывается:

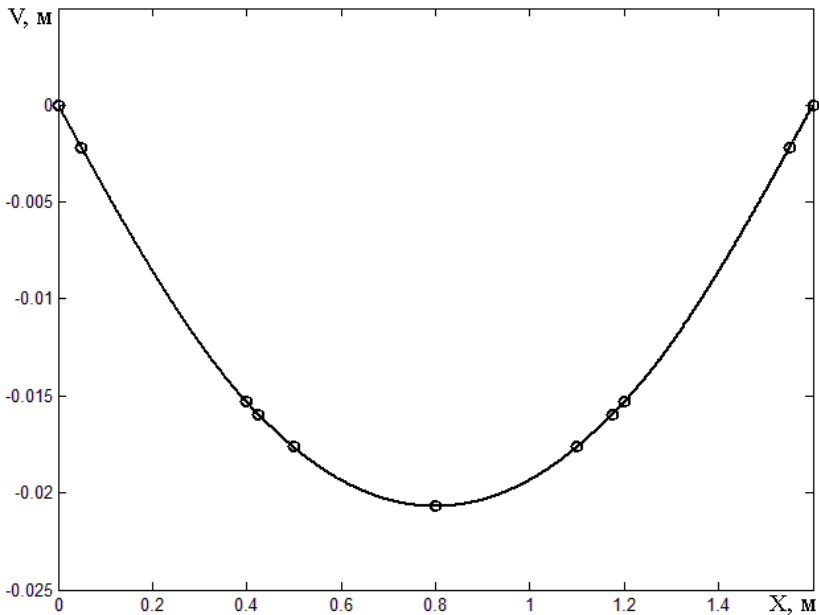
```
% Вычисление значений интерполяционной функции
J=0;% Порядковый номер значения интерполяционной функции
h=0.001;% Шаг плотности вычисления промежуточных точек h
for(I=2:N)
    for(v=X(I-1):h:X(I))
        J=J+1;
```

```

V=Y(I-1)*((X(I)-v)^2)*(2*(v-X(I-1))+D(I))/(D(I)^3)+Y(I)*((v-X(I-1))^2)*(2*(X(I)-v)+D(I))/(D(I)^3)+PY(I-1)*((X(I)-v)^2)*(v-X(I-1))/(D(I)^2)-PY(I)*((v-X(I-1))^2)*(X(I)-v)/(D(I)^2);
x(J)=v;
y(J)=V;
end;
end;
x=x';y=y';
% График сплайна
plot(x,y,X,Y,'ok');

```

Полученная модель прогиба представлена на рис. 7.



**Рис. 7. Функция прогиба балки
передней оси автомобиля «КАМАЗ – 65115»**

4.5. Пример определения функции перемещений сечений стержня с линейным законом изменения жесткости на растяжение-сжатие

Найдем, в качестве примера, функцию перемещений сечений стержня с линейным законом изменения жесткости на растяжение-сжатие ($\gamma = 1$), нагруженного распределенной внешней продольной нагрузкой постоянной интенсивности p и сосредоточенной осевой силой N (см. рис. 8). Стержень закреплен в левом конце при $x = a$, сосредоточенная сила N приложена в правом конце стержня при $x = b$.

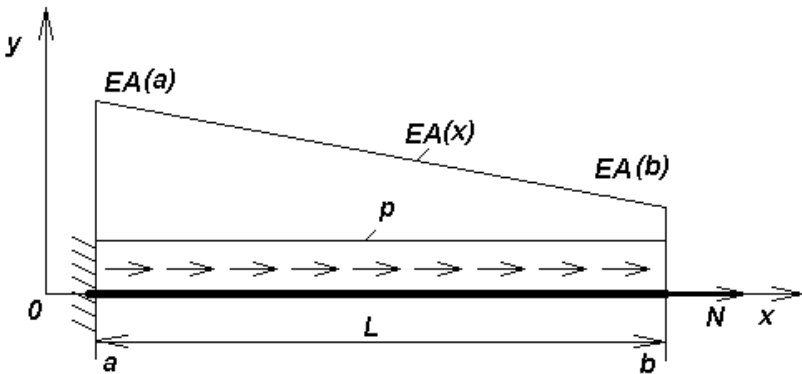


Рис. 8. Схема стержня с действующей нагрузкой, график функции жесткости на растяжение-сжатие

Производная функции перемещений стержня в общем виде определяется выражением (1.26):

$$u'(x) = -\frac{pB^{1-\gamma}(x)}{k} + \frac{c_1}{B^\gamma(x)}, \quad \forall x \in (a, b). \quad (4.34)$$

При $\gamma = 1$ функция жесткости аппроксимируется линейной (аффинной) функцией $B(x)$. В этом случае выражение для производной функции перемещений (4.34) запишется в следующем виде:

$$u'(x) = -\frac{p}{k} + \frac{c_1}{B(x)}. \quad (4.35)$$

После понижения порядка выражения (4.35) получим функцию перемещений для стержней с линейным изменением функции жесткости:

$$u(x) = -\frac{pB(x)}{k^2} + c_1 \frac{\ln(B(x))}{k} + c_2. \quad (4.36)$$

Запишем краевые условия для рассматриваемого стержня. При значении $x = a$ имеем условия закрепления начального сечения стержня:

$$u(a) = 0. \quad (4.37)$$

При значении $x = b$ имеем условие незакрепленного сечения стержня, нагруженного сосредоточенной осевой силой N :

$$B'(b)u'(b) = N. \quad (4.38)$$

Далее, на основании выражений краевых условий (4.37), (4.38) составляем систему линейных алгебраических уравнений для нахождения постоянных интегрирования c_1 и c_2 . Для этого подставим выражение для функции перемещений (4.36) в краевые условия (4.37), (4.38) и получим следующую систему:

$$\begin{cases} -\frac{pB(a)}{k^2} + c_1 \frac{\ln(B(a))}{k} + c_2 = 0 \\ -\frac{pB(b)}{k} + c_1 = N \end{cases}. \quad (4.39)$$

Система уравнений (4.39) для нахождения постоянных интегрирования получилась с треугольной матрицей. Из этой системы получаются выражения для искомым постоянных интегрирования:

$$\begin{aligned} c_1 &= \frac{pB(b)}{k} + N; \\ c_2 &= \frac{p[B(a) - B(b)\ln(B(a))]}{k^2} - \frac{\ln(B(a))}{k} N. \end{aligned} \quad (4.40)$$

Подставляя формулы для постоянных интегрирования (4.40) в выражение для функции перемещений (4.36), получим:

$$u(x) = \frac{p}{k^2} \left[B(b) \ln\left(\frac{B(x)}{B(a)}\right) - B(x) + B(a) \right] + \frac{N}{k} \ln\left(\frac{B(x)}{B(a)}\right). \quad (4.41)$$

Учитывая, что в данном случае $B(x) = EA(x)$ и $k = [B(b) - B(a)]/L$, выражение функции перемещений стержня (4.41) можно переписать в следующем виде, удобном для инженерных расчетов:

$$u(x) = \frac{pL^2}{(EA(b) - EA(a))^2} \left[EA(b) \ln \left(\frac{EA(x)}{EA(a)} \right) - EA(x) + EA(a) \right] + \frac{NL}{EA(b) - EA(a)} \ln \left(\frac{EA(x)}{EA(a)} \right). \quad (4.42)$$

4.6. Пример определения точного решения для консольной балки с линейным законом изменения изгибной жесткости

Найдем, в качестве примера, функцию прогибов консольной балки с линейным законом изменения изгибной жесткости, нагруженной только внешней распределенной нагрузкой постоянной интенсивности (рис. 9).

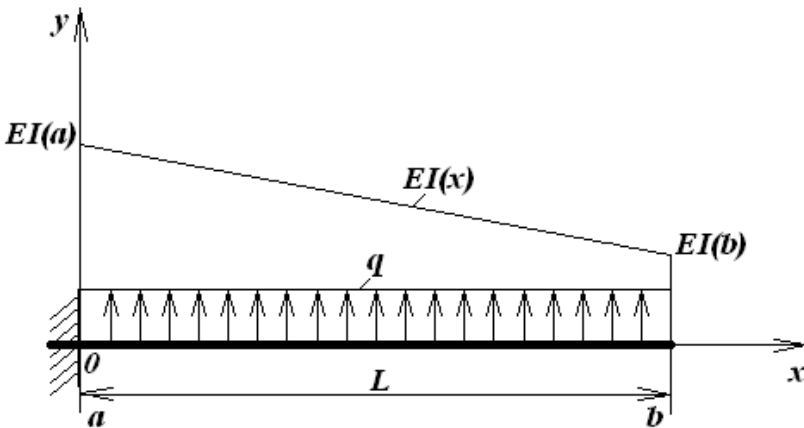


Рис. 9. Схема консольной балки с действующей нагрузкой, график функции изгибной жесткости

Вторая производная функции прогибов балки в общем виде определяется выражением (1.26):

$$v''(x) = q \frac{e^{2-\gamma}(x)}{2k^2} + c_1 \frac{e^{1-\gamma}(x)}{k} + c_2 \frac{1}{e^\gamma(x)}, \quad \forall x \in (a, b). \quad (4.43)$$

Случай при значении $\gamma = 1$ соответствует линейному (аффинному) изменению функции изгибной жесткости балки. В этом случае выражение для второй производной функции прогибов (4.43) запишется в следующем виде:

$$v''(x) = q \frac{e(x)}{2k^2} + C_1 \frac{1}{k} + C_2 \frac{1}{e(x)}. \quad (4.44)$$

После понижения порядка выражения (4.44) получим производную функции прогибов:

$$v'(x) = q \frac{e^2(x)}{4k^3} + C_1 \frac{e(x)}{k^2} + C_2 \frac{\ln(e(x))}{k} + C_3. \quad (4.45)$$

В результате еще одного понижения порядка производной функции прогибов (4.45) получается следующее выражение функции прогибов для балок с линейным изменением функции изгибной жесткости:

$$v(x) = q \frac{e^3(x)}{12k^4} + C_1 \frac{e^2(x)}{2k^3} + C_2 \frac{e(x)[\ln(e(x)) - 1]}{k^2} + C_3 \frac{e(x)}{k} + C_4. \quad (4.46)$$

Запишем краевые условия для рассматриваемой консольной балки со свободным ненагруженным концевым сечением.

При значении $x = a$ имеем условия заделки начального сечения балки:

$$v(a) = 0, \quad v'(a) = 0. \quad (4.47)$$

При значении $x = b$ имеем условия свободного ненагруженного сечения балки. Это условия отсутствия в сечении изгибающего момента M и сосредоточенной поперечной силы F :

$$e^\gamma(b)v''(b) = M = 0, \quad [e^\gamma(b)v''(b)] = F = 0. \quad (4.48)$$

Краевые условия (4.48) можно записать проще:

$$v''(b) = 0, \quad [e^\gamma(b)v''(b)] = 0. \quad (4.49)$$

Далее на основании выражений для краевых условий (4.47) и (4.49) составляем систему линейных алгебраических уравнений для нахождения постоянных интегрирования $C_1, C_2, C_3,$

C_4 . Для этого подставим функцию прогибов (4.46) в краевые условия (4.47), (4.49), получим следующую систему уравнений:

$$\left\{ \begin{array}{l} C_1 \frac{e^2(a)}{2k^3} + C_2 \frac{e(a)[\ln(e(a))-1]}{k^2} + C_3 \frac{e(a)}{k} + C_4 + q \frac{e^3(a)}{12k^4} = 0 \\ C_1 \frac{e(a)}{k^2} + C_2 \frac{\ln(e(a))}{k} + C_3 + q \frac{e^2(a)}{4k^3} = 0 \\ C_1 \frac{1}{k} + C_2 \frac{1}{e(b)} + q \frac{e(b)}{2k^2} = 0 \\ C_1 + q \frac{e(b)}{k} = 0 \end{array} \right. \quad (4.50)$$

Система уравнений (4.50) для нахождения постоянных интегрирования получилась с треугольной матрицей. Из этой системы получают выражения для искомым постоянных интегрирования:

$$\begin{aligned} C_1 &= -q \frac{e(b)}{k}; \quad C_2 = q \frac{e^2(b)}{2k^2}; \\ C_3 &= -\frac{q}{k^3} \left[\frac{e^2(a)}{4} - e(a)e(b) + \frac{e^2(b)\ln(e(a))}{2} \right]; \\ C_4 &= \frac{qe(a)}{2k^4} \left[e^2(b) - e(a)e(b) + \frac{e^2(a)}{3} \right]. \end{aligned} \quad (4.51)$$

Подставляя формулы для постоянных интегрирования (4.51) в выражение для функции прогибов (4.46), получим:

$$\begin{aligned} v(x) &= \frac{q}{k^4} \left\{ \frac{e^3(x)}{12} - \frac{e(b)}{2} e^2(x) + \frac{e^2(b)}{2} e(x)[\ln(e(x))-1] + \right. \\ &\quad \left. + \left[e(a)e(b) - \frac{e^2(a)}{4} - \frac{e^2(b)\ln(e(a))}{2} \right] e(x) + \right. \\ &\quad \left. + \frac{e(a)e^2(b)}{2} + \frac{e^3(a)}{6} - \frac{e^2(a)e(b)}{2} \right\}. \end{aligned} \quad (4.52)$$

В общих случаях могут получаться довольно громоздкие расчетные формулы. Тогда коэффициенты системы для нахождения постоянных интегрирования можно сразу вычислить и полученную систему с числовыми коэффициентами решать

численным методом (например, методом Гаусса). Далее, после подстановки полученных численных значений постоянных интегрирования C_1, C_2, C_3, C_4 в соответствующее общее решение, получается точное частное решение с числовыми коэффициентами.

4.7. Пример проектирования поперечной балки основания грузовой платформы автомобиля на основе точного решения уравнения деформации балки

Описание конструкции грузовой платформы автомобиля и методов расчета. Платформа грузового автомобиля – одна из наиболее металлоемких его конструкций. Выполняется она чаще полунесущей, как это сделано на автомобилях КАМАЗ. На поперечном сечении платформы (рис. 10) показаны основные ее элементы – продольный брус 3 основания, который крепится к раме 1 посредством стремянок 2. К брусу при помощи кронштейна 4 подсоединены поперечные балки 5 основания, связанные между собой (по контуру) стержнями обвязки 7, а в продольном направлении – усилителями пола 9. На образованный таким образом каркас укладываются щиты 8 настила пола. В средней части платформы, в узле соединения поперечной балки 5 с обвязкой 7, крепятся вертикальные стойки 6, которые служат промежуточными опорами бортов 10, соединенных с обвязкой шарнирно, а между собой и стойками – посредством запора. Таким образом, продольные и поперечные элементы основания платформы, изготовленные в основном из тонкостенных стержней открытого профиля, образуют сложную пространственную систему, подкрепленную настилом пола и бортами.

Основными несущими элементами платформы являются поперечные балки 5 основания. От их жесткости, прочности и усталостной долговечности в значительной степени зависит работоспособность платформы в целом. Исследования нагруженности платформ в эксплуатационных условиях показали, что поперечные балки основания подвержены сложному многокомпонентному нагружению. Так, вертикальные нагрузки от пере-

возимого автомобилем груза и сил инерции, действуя через настил пола, изгибают поперечные балки в вертикальной плоскости. При движении автомобиля по неровным дорогам, от перекосов автомобиля, поперечные балки основания платформы подвергаются переменному закручиванию и изгибу в горизонтальной плоскости. Поперечные и продольные силы инерции воспринимаются в основном кронштейнами крепления 4 поперечных балок 5 к продольным брусам 3 основания, а при перевозке сыпучих грузов – также бортами с обвязкой.

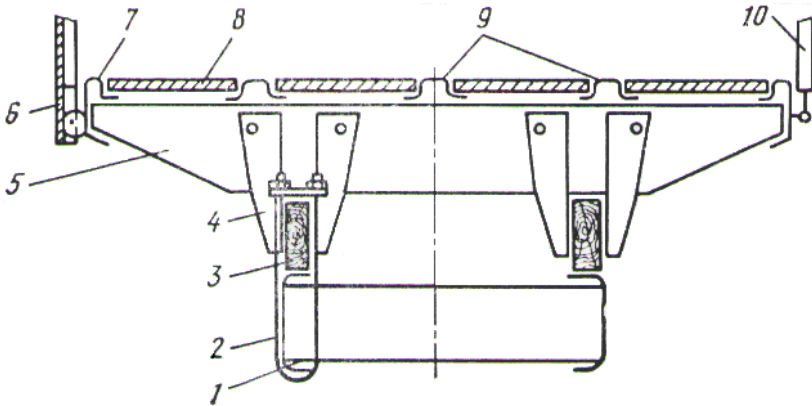


Рис. 10. Схема поперечного сечения бортовой платформы автомобилей КАМАЗ:

- 1 – рама; 2 – стремянка; 3 – брус; 4 – кронштейн; 5 – балка; 6 – стойка;
7 – обвязка; 8 – щит настила пола; 9 – усилитель пола; 10 – борт

Поперечные балки чаще бывают швеллерного сечения и изготавливаются штамповкой из стального листа с переменным моментом инерции $I(x)$ сечения с координатой x на консольных участках (рис. 11).

Практикой эксплуатации установлено, что если прогибы консольных участков балки под действием распределенной по ее длине нагрузки, в 1,5 раза превышающей номинальную нагрузку, приходящуюся на балку от перевозимого груза, не превышают $l_k / 400$, то балка обладает необходимой жесткостью. При этом, как правило, балка имеет и требуемую прочность. Усталостная долговечность балки при этом обеспечивается пра-

вильным конструированием кронштейнов крепления балки к продольным брусам платформы. Кронштейны должны воспринимать, не разрушаясь, вертикальные, продольные и поперечные нагрузки, но не стеснять деформацию балки при кручении, возникающем от перекосов автомобиля в случае движения по неровным дорогам. При проектировании длина балки и расстояние между опорами на продольных брусках обычно заданы (определяются поперечными габаритами автомобиля и конструкцией рамы его шасси). Высоты сечений балки $H(x)$ также ограничены требованиями по погрузочной высоте пола платформы и обеспечению свободного перемещения колес автомобиля при движении по неровным дорогам. Поэтому оптимизация балки по металлоемкости может вестись за счет варьирования толщины профиля $\delta = (H - h)/2$. Нагрузка q , действующая на балку в вертикальной плоскости, при расчетах принимается равномерно распределенной по ее длине и определяется как отношение веса перевозимого груза Q к произведению числа балок n на длину одной такой балки $(l_n + 2l_k)$. Расчетная схема поперечной балки и функция жесткости представлены на рис. 11.

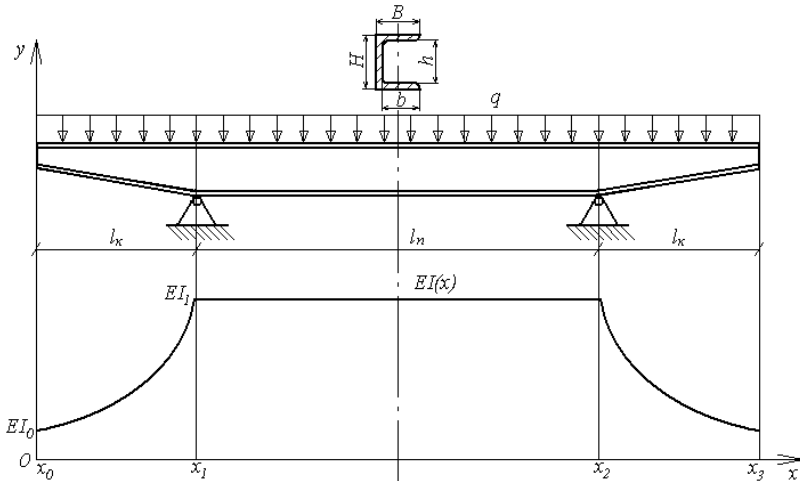


Рис. 11. Схема нагружения поперечной балки и график функции изгибной жесткости

Обычно при проектировании такой балки функцию ее изгибной жесткости заменяют на эквивалентную функцию жесткости, постоянную по всей длине балки, что ведет к значительным погрешностям в оценке напряженно-деформированного состояния и, следовательно, повышению металлоемкости. Для повышения точности анализа рассматривают более сложную конечно-элементную модель балки с кусочно-постоянной функцией жесткости. В этом случае большое количество параметров не позволяет оценить их влияние на общую картину деформации балки. Но такая оценка необходима при проектных расчетах и оптимизации.

В данном примере рассмотрим метод проектирования этой балки, избавленный от вышеназванных недостатков. Здесь необходимо рассмотреть дифференциальное уравнение поперечного изгиба балки (1.22), которое для данного случая (рис. 11) имеет вид:

$$[EI(x)v''(x)]'' = q, \quad \forall x \in (x_{i-1}, x_i), \quad i = 1, 2, 3, \quad (4.53)$$

где $v(x)$ – функция прогибов оси балки; $EI(x)$ – функция изгибной жесткости балки; q – поперечная внешняя распределенная нагрузка, действующая на ось балки Ox в плоскости Oxy и постоянная по ее длине; x_{i-1} , x_i – соответственно координаты начала и конца рассматриваемого участка балки по оси Ox ; оси Oy и Oz направлены соответственно с главными центральными осями инерции поперечного сечения балки.

При проектировании необходимо располагать функцией прогибов проектируемой конструкции, являющейся решением уравнения (4.53). В разделе 1.6 рассмотрены вопросы получения точных общих решений уравнения (4.53), в которых функция изгибной жесткости аппроксимируется специальным степенным законом в виде:

$$EI(x) \cong e^{\gamma}(x), \quad \forall x \in [x_{i-1}, x_i], \quad (4.54)$$

где $e(x) = [(x - x_{i-1})e(x_i) - (x - x_i)e(x_{i-1})] / (x_i - x_{i-1})$ – линейная (аффинная) функция; $e(x_{i-1}) = \sqrt[\gamma]{EI(x_{i-1})}$, $e(x_i) = \sqrt[\gamma]{EI(x_i)}$ – значения функции $e(x)$ для начального и конечного сечений i -го участка

балки, соответственно; γ_i – показатель степени для i -го участка балки.

Решения уравнения (4.53) основываются на технической теории изгиба балок, согласно которой вместо изгиба балки рассматривается изгиб ее оси Ox . Та же теория используется при проектировании поперечных балок постоянной жесткости и при анализе балочных конструкций методом конечных элементов.

Построение функции прогибов поперечной балки платформы. Разобьем балку на три участка в соответствии с функцией жесткости, где 1-й и 3-й участки являются консольными. Практическими расчетами выявлено, что показатель степени γ аппроксимирующей функции для консольных участков принимает дробные значения и в силу симметрии $\gamma_1 = \gamma_3$, а в среднем участке равняется нулю, так как жесткость постоянна по длине участка. Ввиду этого индексы при показателе степени опустим.

Для получения функции прогибов нужно решать уравнение (4.53) для всех 3-х участков, соответственно аппроксимировав функцию жесткости специальным степенным законом (4.54).

Случай при значении $\gamma = 0$ соответствует постоянной жесткости балки. В этом случае решение уравнения деформации балки запишется в следующем виде:

$$v(x) = \int \int \int \int \frac{q}{EI} (dx)^4 = q \frac{x^4}{24EI} + C_1 \frac{x^3}{6} + C_2 \frac{x^2}{2} + C_3 x + C_4. \quad (4.55)$$

Рассмотрим получение такого решения уравнения (4.54) для любых показателей степени γ . В результате последовательного понижения выражения для второй производной функции прогибов балки переменной жесткости (1.26, 4.43) получаются следующие выражения:

$$v''(x) = q \frac{e^{2-\gamma}(x)}{2k^2} + C_1 \frac{e^{1-\gamma}(x)}{k} + C_2 \frac{1}{e^\gamma(x)}; \quad (4.56)$$

$$v'(x) = q \frac{e^{3-\gamma}(x)}{2(3-\gamma)k^3} + C_1 \frac{e^{2-\gamma}(x)}{(2-\gamma)k^2} + C_2 \frac{e^{1-\gamma}(x)}{(1-\gamma)k} + C_3; \quad (4.57)$$

$$v(x) = q \frac{e^{4-\gamma}(x)}{2(3-\gamma)(4-\gamma)k^4} + C_1 \frac{e^{3-\gamma}(x)}{(2-\gamma)(3-\gamma)k^3} +$$

$$+ C_2 \frac{e^{2-\gamma}(x)}{(1-\gamma)(2-\gamma)k^2} + C_3 \frac{e(x)}{k} + C_4. \quad (4.58)$$

Запишем функцию прогибов для участков.

Функция прогибов для первого участка (консоль, $x \in [x_0, x_1]$) (4.58):

$$v_1(x) = q \frac{e_1^{4-\gamma}(x)}{2(3-\gamma)(4-\gamma)k_1^4} + C_{1,1} \frac{e_1^{3-\gamma}(x)}{(2-\gamma)(3-\gamma)k_1^3} + \\ + C_{2,1} \frac{e_1^{2-\gamma}(x)}{(1-\gamma)(2-\gamma)k_1^2} + C_{3,1} \frac{e_1(x)}{k_1} + C_{4,1}, \quad (4.59)$$

где $C_{1,i}$, $C_{2,i}$, $C_{3,i}$, $C_{4,i}$ – неизвестные константы интегрирования i -го участка; $k_i = (e_i - e_{i-1})/(x_i - x_{i-1})$ – постоянный коэффициент i -го участка.

Функция прогибов для второго участка (пролет, $x \in [x_1, x_2]$) (4.55):

$$v_2(x) = q \frac{x^4}{24EI} + C_{1,2} \frac{x^3}{6} + C_{2,2} \frac{x^2}{2} + C_{3,2}x + C_{4,2}. \quad (4.60)$$

Функция прогибов для третьего участка (консоль, $x \in [x_2, x_3]$) (4.58):

$$v_3(x) = q \frac{e_3^{4-\gamma}(x)}{2(3-\gamma)(4-\gamma)k_3^4} + C_{1,3} \frac{e_3^{3-\gamma}(x)}{(2-\gamma)(3-\gamma)k_3^3} + \\ + C_{2,3} \frac{e_3^{2-\gamma}(x)}{(1-\gamma)(2-\gamma)k_3^2} + C_{3,3} \frac{e_3(x)}{k_3} + C_{4,3}. \quad (4.61)$$

Неизвестные константы интегрирования находятся из граничных условий участков балки в аналитическом виде. Структура и симметричность получаемой системы уравнений позволяет рассмотреть каждый участок по отдельности, введя дополнительную неизвестную $p_1 = p(x_1)$ – первую производную функции прогибов в шарнирной опоре при x_1 . Учитывая симметричность конструкции, получим $p_2 = p(x_2) = -p_1$, и рассматривать третий, консольный, участок нет необходимости. Поэтому индексы при постоянных γ , k можно опустить, а для функции жесткости ввести новый смысл индексов: $e_i = e(x_i)$.

Запишем граничные условия для первого консольного участка:

при $x = x_0$ (свободная консоль)

отсутствие сосредоточенной перерезывающей силы:

$$\left[e^\gamma(x_0) v_1''(x_0) \right] = q \frac{e_0}{k} + C_{1,1} = 0; \quad (4.62)$$

отсутствие сосредоточенного изгибающего момента:

$$v_1''(x_0) = q \frac{e_0^2}{2k^2} + C_{1,1} \frac{e_0}{k} + C_{2,1} = 0; \quad (4.63)$$

при $x = x_1$ (шарнирная опора)

опирание на производную функции прогибов:

$$v_1'(x_1) = q \frac{e_1^{3-\gamma}}{2(3-\gamma)k^3} + C_{1,1} \frac{e_1^{2-\gamma}}{(2-\gamma)k^2} + C_{2,1} \frac{e_1^{1-\gamma}}{(1-\gamma)k} + C_{3,1} = p_1; \quad (4.64)$$

исключение прогиба в опоре:

$$v_1(x_1) = q \frac{e_1^{4-\gamma}}{2(3-\gamma)(4-\gamma)k^4} + C_{1,1} \frac{e_1^{3-\gamma}}{(2-\gamma)(3-\gamma)k^3} + C_{2,1} \frac{e_1^{2-\gamma}}{(1-\gamma)(2-\gamma)k^2} + C_{3,1} \frac{e_1}{k} + C_{4,1} = 0. \quad (4.65)$$

Нетрудно видеть, что система уравнений (4.62) – (4.65) для нахождения постоянных интегрирования получилась с треугольной матрицей. Из этой системы несложно получаются выражения для искоемых постоянных интегрирования:

$$\begin{aligned} C_{1,1} &= -q \frac{e_0}{k}; \quad C_{2,1} = -q \frac{e_0^2}{2k^2}; \\ C_{3,1} &= p_1 - \frac{q}{k^3} \left[\frac{e_1^{3-\gamma}}{2(3-\gamma)} - \frac{e_0 e_1^{2-\gamma}}{(2-\gamma)} + \frac{e_0^2 e_1^{1-\gamma}}{2(1-\gamma)} \right]; \\ C_{4,1} &= -p_1 \frac{e_1}{k} + \frac{q}{k^4} \left[\frac{e_1^{4-\gamma}}{2(4-\gamma)} - \frac{e_0 e_1^{3-\gamma}}{(3-\gamma)} + \frac{e_0^2 e_1^{2-\gamma}}{2(2-\gamma)} \right]. \end{aligned} \quad (4.66)$$

Подставляя формулы для постоянных интегрирования (4.66) в выражение для функции прогибов консоли (4.59), получим:

$$v_1(x) = q \frac{e^{4-\gamma}(x)}{2(3-\gamma)(4-\gamma)k^4} - q \frac{e_0 e^{3-\gamma}(x)}{(2-\gamma)(3-\gamma)k^4} + q \frac{e_0^2 e^{2-\gamma}(x)}{2(1-\gamma)(2-\gamma)k^4} +$$

$$\begin{aligned}
& + \left\{ \frac{p_1}{k} - \frac{q}{k^4} \left[\frac{e_1^{3-\gamma}}{2(3-\gamma)} - \frac{e_0 e_1^{2-\gamma}}{(2-\gamma)} + \frac{e_0^2 e_1^{1-\gamma}}{2(1-\gamma)} \right] \right\} e(x) - \\
& - p_1 \frac{e_1}{k} + \frac{q}{k^4} \left[\frac{e_1^{4-\gamma}}{2(4-\gamma)} - \frac{e_0 e_1^{3-\gamma}}{(3-\gamma)} + \frac{e_0^2 e_1^{2-\gamma}}{2(2-\gamma)} \right]. \quad (4.67)
\end{aligned}$$

В выражении (4.67) неизвестным является значение производной в шарнирной опоре p_1 , для его определения необходимо рассмотреть второй участок (пролет) балки.

Запишем граничные условия для второго участка (пролета):
при $x = x_1$ (шарнирная опора)

исключение прогиба в опоре:

$$v_2(x_1) = q \frac{x_1^4}{24EI} + C_{1,2} \frac{x_1^3}{6} + C_{2,2} \frac{x_1^2}{2} + C_{3,2} x_1 + C_{4,2} = 0, \quad (4.68)$$

где $EI = EI(x) = const \quad \forall x \in [x_1, x_2]$, отметим, что $EI = e_1^\gamma$;
опиание на производную функции прогибов:

$$v_2'(x_1) = q \frac{x_1^3}{6EI} + C_{1,2} \frac{x_1^2}{2} + C_{2,2} x_1 + C_{3,2} = p_1; \quad (4.69)$$

при $x = x_2$ (шарнирная опора)

исключение прогиба в опоре:

$$v_2(x_2) = q \frac{x_2^4}{24EI} + C_{1,2} \frac{x_2^3}{6} + C_{2,2} \frac{x_2^2}{2} + C_{3,2} x_2 + C_{4,2} = 0, \quad (4.70)$$

опиание на производную функции прогибов:

$$v_2'(x_2) = q \frac{x_2^3}{6EI} + C_{1,2} \frac{x_2^2}{2} + C_{2,2} x_2 + C_{3,2} = p_2 = -p_1. \quad (4.71)$$

Для аналитического решения линейной системы уравнений (4.68) – (4.71) применим среду символьной математики MathCAD 2000 Professional, найденные в ней искомые постоянные интегрирования запишутся в виде:

$$\begin{aligned}
C_{1,2} &= -q \frac{x_1 + x_2}{2EI}; \quad C_{2,2} = -\frac{2p_1}{x_2 - x_1} + q \frac{[(x_1 + x_2)^2 + 2x_1 x_2]}{12EI}; \\
C_{3,2} &= \frac{p_1(x_1 + x_2)}{x_2 - x_1} - q \frac{(x_1 + x_2)x_1 x_2}{12EI};
\end{aligned}$$

$$C_{4.2} = -\frac{p_1 x_1 x_2}{x_2 - x_1} + q \frac{x_1^2 x_2^2}{24EI}. \quad (4.72)$$

Подставляя формулы для постоянных интегрирования (4.72) в выражение для функции прогибов пролета (4.60), получим:

$$v_2(x) = q \frac{x^4}{24EI} - q \frac{(x_1 + x_2)x^3}{12EI} + \left\{ -\frac{p_1}{x_2 - x_1} + q \frac{[(x_1 + x_2)^2 + 2x_1 x_2]}{24EI} \right\} x^2 + \left[\frac{p_1(x_1 + x_2)}{x_2 - x_1} - q \frac{(x_1 + x_2)x_1 x_2}{12EI} \right] x - \frac{p_1 x_1 x_2}{x_2 - x_1} + q \frac{x_1^2 x_2^2}{24EI}. \quad (4.73)$$

Неизвестную производную p_1 найдем из условия неразрывности функции моментов $M(x) = EI(x)v''(x)$ на шарнирной опоре в сечении x_1 :

$$EI(x_1 - 0)v_1''(x_1 - 0) = EI(x_1 + 0)v_1''(x_1 + 0). \quad (4.74)$$

Подставляя в условие (4.74) вторые производные от выражений (4.59) (см. (4.56)) и (4.60) и формулы для постоянных интегрирования (4.66) и (4.72), получим:

$$q \frac{e_1^2}{2k^2} - q \frac{e_0 e_1}{k^2} + q \frac{e_0^2}{2k^2} \equiv q \frac{(x_1 - x_0)^2}{2} = EI \left\{ q \frac{x^2}{2EI} - q \frac{(x_1 + x_2)x_1}{2EI} - \frac{2p_1}{x_2 - x_1} + q \frac{[(x_1 + x_2)^2 + 2x_1 x_2]}{12EI} \right\}. \quad (4.75)$$

Из выражения (4.75), после упрощений, получим искомую неизвестную:

$$p_1 = \frac{q}{24EI} \left[(x_2 - x_1)^3 - 6(x_2 - x_1)(x_1 - x_0)^2 \right] \\ \text{или } p_1 = \frac{q e_1^{-\gamma}}{24} (l_n^3 - 6l_n l_k^2). \quad (4.76)$$

Расчетные соотношения проектного метода. Как уже говорилось, проектный расчет поперечных балок сводится к расчету на изгибную жесткость. При этом удовлетворяется и их прочность. Целью такого расчета является определение толщины профиля δ , значения которой должны лежать в интервале от 4 до 6 мм, так как при меньших значениях трудно обеспечить

местную прочность при креплении балок с другими элементами платформы, а при больших – возникают технологические трудности.

Максимальный прогиб, как правило, возникает на краю консоли при x_0 и не должен превышать $l_k/400$. Подставим в выражение для функции прогиба консоли (4.67) $x = x_0 \Rightarrow e(x) = e_0$, упростим и получим выражение для максимального прогиба:

$$v_{1,\max}(x_0) = \frac{q}{2k^4} \left[\frac{6e_0^{4-\gamma}}{(1-\gamma)(2-\gamma)(3-\gamma)(4-\gamma)} + \frac{e_1^{4-\gamma}}{(4-\gamma)} - \frac{3e_0e_1^{3-\gamma}}{(3-\gamma)} + \frac{3e_0^2e_1^{2-\gamma}}{(2-\gamma)} - \frac{e_0^3e_1^{1-\gamma}}{(1-\gamma)} \right] - \frac{qe_1^{-\gamma}}{24} (I_n^3 l_k - 6I_n l_k^3) \leq \frac{l_k}{400}, \quad (4.77)$$

где $q = n_d Q / [n(l_n + 2l_k)]$ – распределенная внешняя нагрузка на балку; $n_d = 1,5$ – коэффициент динамичности нагрузки; Q – вес перевозимого груза; n – число поперечных балок платформы; l_n – длина пролета балки между опорами; l_k – длина консоли балки; $k = (e_1 - e_0) / l_k$ – постоянный коэффициент консоли – производная линейной функции $e(x)$, аппроксимирующей по специальному степенному закону (4.54) функцию изгибной жесткости консоли $EI(x)$; $e_0 = \sqrt[\gamma]{EI_0}$; $e_1 = \sqrt[\gamma]{EI_1}$; γ – показатель степени специального степенного закона (4.54); $EI_0 = EI(x_0)$, $EI_1 = EI(x_1)$ – значения изгибной жесткости на краю и в конце (на опоре) консоли соответственно.

Функция жесткости консоли швеллерного сечения имеет вид (обозначения размеров поперечного сечения балки с рис. 11):

$$EI(x) = \frac{E}{12} (BH^3(x) - bh^3(x)) = \frac{E}{12} [BH^3(x) - b(H(x) - 2(B-b))^3] \quad \forall x \in [x_0, x_1], \quad (4.78)$$

где $H(x) = [(x - x_0)H(x_1) - (x - x_1)H(x_0)] / l_k$; $h(x) = H(x) - 2(B - b)$; момент инерции $I(x)$ берется относительно главной центральной оси Oz (индекс опущен).

Проектный расчет следует вести подбором толщины профиля $\delta = (H - h)/2 = B - b$ из условия (4.77): задаются стандартные значения δ из интервала от 0,004 до 0,006 м; по формуле (4.78) определяются значения жесткостей в начале, середине и конце консольного участка EI_0 , EI_{cp} , EI_1 ; по найденным значениям функция жесткости аппроксимируется специальным степенным законом (раздел 1.6) и определяются значения γ , e_0 , e_1 ; найденные значения подставляются в выражение (4.77), где определяется максимальный прогиб и сравнивается с допусковым. Все вычисления следует проводить в размерностях системы [СИ].

Проектный расчет поперечной балки грузовой платформы «КАМАЗ – 53215». Исходными данными для расчета поперечной балки грузовой платформы автомобиля «КАМАЗ» являются (размерность [СИ]):

длина консоли балки $l_k = 0,815$ м; длина пролета балки между опорами $l_n = 0,87$ м; высоты сечений на краю и в конце консоли балки $H_0 = H(x_0) = 0,05$ м и $H_1 = H(x_1) = 0,16$ м, соответственно; ширина полки швеллерного сечения $B = 0,06$ м; модуль упругости материала балки $E = 2 \cdot 10^{11}$ Па; вес перевозимого груза $Q = 110000$ Н; число поперечных балок платформы $n = 7$; коэффициент динамичности нагрузки $n_d = 1,5$.

Зададим толщину профиля сечения балки: $\delta = 0,004$ м.

Определим размер $b = B - \delta = 0,056$ м; высоту в среднем сечении консоли по формуле $H_{cp} = (H_0 + H_1)/2 = 0,105$ м; внешнюю распределенную нагрузку (с учетом знака) $q = -n_d Q / [n(l_n + 2l_k)] = -9,429 \cdot 10^3$ Н/м и значения изгибной жесткости на краю, в середине и в конце консоли балки:

$$EI_0 = [BH_0^3 - b(H_0 - 2(B - b))^3]E/12 = 5,585 \cdot 10^4 \text{ Н} \cdot \text{м}^2,$$

$$EI_{cp} = [BH_{cp}^3 - b(H_{cp} - 2(B - b))^3]E/12 = 3,058 \cdot 10^5 \text{ Н} \cdot \text{м}^2,$$

$$EI_1 = [BH_1^3 - b(H_1 - 2(B - b))^3]E/12 = 8,183 \cdot 10^5 \text{ Н} \cdot \text{м}^2, \text{ соответ-}$$

ственно.

Допускаемый прогиб составит $[v] = l_k / 400 = 2,037 \cdot 10^{-3}$ м.

Аппроксимируем функцию изгибной жесткости консоли балки специальным степенным законом (4.54) первым вариантом (раздел 1.6). Для этого определим степенной коэффициент $c = \ln(EI_0 / EI_{cp}) / \ln(EI_1 / EI_{cp}) = -1,727$ и зададим значение начального приближения переменной $z_{(0)} = 3$. Итерационный процесс будем вести по формуле $z_{(i)} = 2 - z_{(i-1)}^c$ с точностью до $\varepsilon = 0,001$, то есть пока не выполнится условие сходимости итераций $|z_{(i)} - z_{(i-1)}| < \varepsilon$:

$$z_{(1)} = 2 - z_{(0)}^c = 1,85 \text{ и } |z_{(1)} - z_{(0)}| = 1,15 > \varepsilon;$$

$$z_{(2)} = 2 - z_{(1)}^c = 1,654 \text{ и } |z_{(2)} - z_{(1)}| = 0,196 > \varepsilon;$$

...;

$$z_{(10)} = 2 - z_{(9)}^c = 1,509 \text{ и } |z_{(10)} - z_{(9)}| = 0,001 = \varepsilon;$$

$z_{(11)} = 2 - z_{(10)}^c = 1,509$ и $|z_{(11)} - z_{(10)}| = 0 < \varepsilon$ – выполнение условия сходимости на 11-й итерации.

Отсюда $z = z_{(11)} = 1,509$ и показатель степени $\gamma = \ln(EI_1 / EI_{cp}) / \ln(z) \approx 2,4$. Определим значения линейной функции в начале и конце консоли балки $e_0 = \sqrt[\gamma]{EI_0} = 95,045$ и $e_1 = \sqrt[\gamma]{EI_1} = 290,882$; постоянный коэффициент консоли $k = (e_1 - e_0) / l_k = 240,29$.

По формуле (4.77) определим максимальный прогиб поперечной балки $v_{\max} = v_{1,\max} = -2,081 \cdot 10^{-3}$ м (знак минус означает, что прогиб направлен вниз по оси Oy). Сравним его с допускаемым прогибом $\frac{|v_{\max}| - [v]}{[v]} 100\% = 2,14 < 3\%$, что допустимо.

Таким образом, для изготовления балок следует выбрать стальной лист толщиной $\delta = 4$ мм, что обеспечивает рациональное расходование материала.

Изложенный метод проектного расчета поперечных балок основания платформ грузовых автомобилей позволяет создавать

платформы рациональные по жесткости, прочности, ресурсу и металлоемкости. Полученные выражения не ограничиваются принятым проектным параметром – толщины профиля δ . Выражения для функции прогибов консоли (4.67) и пролета (4.73) с учетом (4.76) позволяют проводить точный анализ и оптимизацию поперечных балок по любым входящим в эти выражения параметрам. По примеру получения данного метода, основанного на точных решениях уравнения деформации балки, можно получить проектные методы любых других балочных конструкций.

На рис. 12 представлены результаты расчета этой же балки как трехмерного (объемного) тела в системе конечно-элементного анализа (CAE-системе) ANSYS в виде ее деформированного состояния, цветом согласно шкале показаны величины прогибов.

При расчете использована сетка, состоящая из 231-го гексагонального объемного конечного элемента. Для сравнения, в аналитическом расчете использовано всего 3 участка (балочных КЭ), что само собой говорит о вычислительных затратах! Распределенная нагрузка приложена к верхнему угловому ребру балки. Максимальный прогиб составляет $\sim 2,1$ мм и направлен против вертикальной оси, что показывает совпадение результатов с точным аналитическим решением до сотых долей мм и подтверждает высокую точность определения напряженно-деформированного состояния. Также критерием правильности полученных результатов служит их симметричность относительно центра балки, так как сама балка и приложенная к ней нагрузка симметричны. Чтобы получить достоверные результаты, пришлось верифицировать (подтверждать) конечно-элементную модель на точном аналитическом решении дифференциального уравнения деформации балки и на реальной конструкции. Поэтому перед расчетом необходимо иметь представление о том, как ведет себя конструкция под нагрузкой, какими свободами перемещений она должна обладать, а какие следует ограничить, как приложить нагрузку и с каким значением, какой примерно результат необходимо ожидать от расчета. Это составляет теоретическое осмысление конструкции.

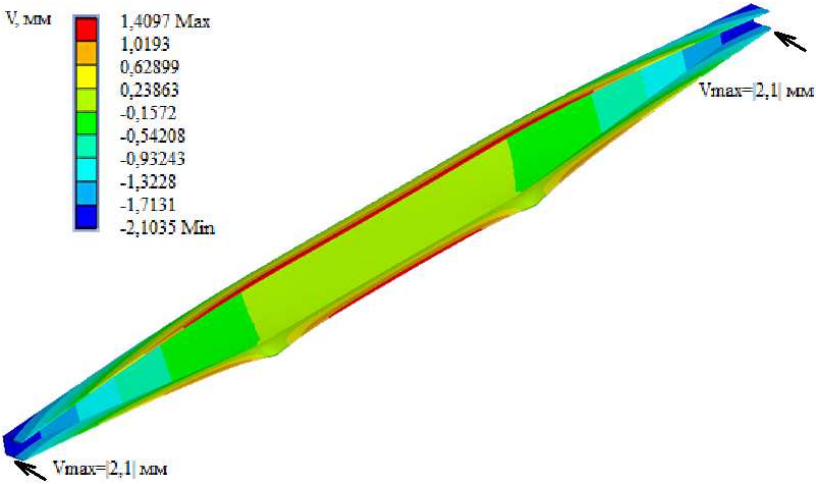


Рис. 12. Деформированное состояние балки

4.8. Вопросы для самопроверки

1. Что такое стержень?
2. На какие виды деформации работают стержни?
3. Как определяется жесткость стержня при растяжении-сжатии?
4. Какие параметры отыскиваются в задаче анализа стержня, работающего на растяжение-сжатие?
5. Каким методом решается задача анализа стержня?
6. Как задаются функции жесткости стержня и внешней распределенной нагрузки?
7. Что представляет собой решение дифференциального уравнения стержня постоянной жесткости на растяжение-сжатие?
8. Что представляет собой решение дифференциального уравнения стержня постоянной жесткости на растяжение-сжатие, разложенное на базисные функции?
9. Как называется матрица системы уравнений для нахождения искомым перемещений стержня?
10. Как называется вектор правых частей системы уравнений для нахождения искомым перемещений стержня?

11. *Какие параметры отыскиваются в задаче оптимального проектирования стержня?*
12. *Какие дополнительные условия используются при оптимальном проектировании стержня?*
13. *Как формулируется условие перераспределения жесткости стержня?*
14. *Как формулируется условие равнонапряженности?*
15. *Как выражаются напряжения?*
16. *Каким методом учитываются дополнительные условия?*
17. *Как получается система уравнений для решения задачи оптимального проектирования стержня?*
18. *Почему непосредственно не решают систему уравнений задачи оптимального проектирования стержня?*
19. *Каким методом решается задача оптимального проектирования стержня?*
20. *Как ускорить поиск решения градиентным методом задачи оптимального проектирования?*
21. *Что такое обвод?*
22. *Что является исходными данными при моделировании обводов?*
23. *Как влияет функция жесткости сплайна при моделировании обводов?*
24. *Каким методом определяются параметры сетки?*
25. *Что требуется определить для задания обводов кубическим сплайном?*
26. *Какой геометрический смысл имеют неизвестные коэффициенты сплайна?*
27. *Каким методом решается система уравнений для нахождения неизвестных коэффициентов сплайна?*
28. *Сколько типов краевых условий может быть у кубического сплайна?*
29. *Какой тип краевых условий используется в случае, когда неизвестны ни первая, ни вторая производные непериодического (незамкнутого) обвода на краях?*
30. *Что является результатом решения задачи моделирования обвода?*
31. *Что такое балка?*

32. *Какие исходные данные необходимы для анализа балки МКЭ на изгиб?*
33. *Как осуществляется разбивка на конечные элементы?*
34. *Как свести сосредоточенный момент к распределенной внешней поперечной нагрузке?*
35. *Какие краевые условия (условия закрепления) возможны у балок?*
36. *Сколько степеней свободы (перемещений) имеет узел балочного конечного элемента?*
37. *Как учитываются краевые условия (условия закрепления) в системе уравнений МКЭ?*
38. *Каким методом решается система уравнений МКЭ?*
39. *Какие неизвестные определяются из решения системы уравнений МКЭ?*
40. *Как строится функция прогибов балки?*

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Ануфриев И.Е. MATLAB 7 / И.Е.Ануфриев, А.Б.Смирнов, Е.Н.Смирнова. – СПб.: БХВ-Петербург, 2005. – 1104 с.
2. Асанов А.З. Моделирование и анализ динамических систем: учеб. пособие. – Набережные Челны: Изд-во Камского гос. политехн. ин-та, 2004. – 152 с.
3. Березин Б.И. Начальный курс С и С++ / Б.И.Березин, С.Б.Березин. – М.: Диалог – МИФИ, 2001. – 288 с.
4. Голованов А.И. Современные конечно-элементные модели и методы исследования тонкостенных конструкций / А.И. Голованов, А.В. Песошин, О.Н. Тюленева. – Казань: Казан. гос. ун-т им. В.И. Ульянова-Ленина, 2005. – 442 с.
5. Гончаров П.С. NX для конструктора-машиностроителя / П.С. Гончаров, М.Ю. Ельцов, С.Б. Коршиков [и др.]. – М.: ДМК Пресс, 2010. – 504 с.
6. Гребеньков О.А. Конструкция самолетов: учебник для вузов / О.А. Гребеньков, В.П. Гоголин, А.И. Осокин, В.Ф. Снигирев [и др.]. – Казань: Изд-во КГТУ, 1999. – 320 с.
7. Данилов Ю. Практическое использование NX / Ю.Данилов, И.Артамонов. – М.: ДМК Пресс, 2011. – 332 с.
8. Дементьев Ю.В. САПР в автомобиле- и тракторостроении: учебник для студ. высш. учеб. заведений / Ю.В.Дементьев, Ю.С.Щетинин. – М.: Академия, 2004. – 224 с.
9. Дьяконов В.П. Maple 7: учеб. курс. – СПб.: Питер, 2002. – 672 с.
10. Калиткин Н.Н. Численные методы: учеб. пособие для студ. высш. учеб. заведений. – М.: Наука, 1978. – 512 с.
11. Мельников А.А. Управление техническими объектами автомобилей и тракторов: Системы электроники и автоматики: учеб. пособие для студ. высш. учеб. заведений. – М.: Академия, 2003. – 376 с.
12. Мэтьюз Д.Г. Численные методы. Использование MATLAB / Д.Г.Мэтьюз, К.Д.Финк. – М.: Вильямс, 2001. – 720 с.
13. Мяченков В.И. Расчеты машиностроительных конструкций методом конечных элементов: справочник / В.И.Мяченков, В.П.Мальцев, В.П.Майборода [и др.]. – М.: Машиностроение, 1989. – 520 с.

14. Павловская Т.А. С/С++. Структурное программирование: практикум / Т.А.Павловская, Ю.А.Щупак. – СПб.: Питер, 2002. – 240 с.
15. Постнов В.А. Метод конечных элементов в расчетах судовых конструкций / В.А.Постнов, И.Я.Хархурим. – Л.: Судостроение, 1974. – 344 с.
16. Россоловский А.В. AutoCAD 2000. Настольная книга пользователя. – М.: Нолидж, 2001. – 928 с.
17. Рычков С.П. MSC.visualNASTRAN для Windows. – М.: НТ Пресс, 2004. – 552 с.
18. Садовничий В.А. Теория операторов: учебник для вузов. – 4-е изд., испр. и доп. – М.: Дрофа, 2001. – 384 с.
19. Сайтов И.Х. САПР в машиностроении: учеб. пособие для студентов машиностроительных специальностей очной и заочной формы обучения / И.Х.Сайтов, В.Ф.Снигирев, А.С.Чернявский. – Казань: Изд-во Казан. гос. техн. ун-та, 2001. – 131 с.
20. Сегерлинд Л. Применение метода конечных элементов. – М.: Мир, 1979. – 392 с.
21. Стренг Г. Теория метода конечных элементов / Г.Стренг, Дж.Фикс. – М.: Мир, 1977. – 349 с.
22. Треногин В.А. Функциональный анализ. – М.: Наука, 1980. – 496 с.
23. Херхагер М. Mathcad 2000: полное руководство: пер. с нем. / М.Херхагер, Х.Партоль. – К.: ВHV, 2000. – 416 с.

**Алексей Петрович Павленко
Вячеслав Николаевич Никишин**

**АНАЛИТИЧЕСКИЕ И ЧИСЛЕННЫЕ МЕТОДЫ
ПРОЧНОСТНОГО АНАЛИЗА И ПРОЕКТИРОВАНИЯ
АВТОМОБИЛЬНЫХ КОНСТРУКЦИЙ**

Учебное пособие

Главный редактор *Д.К. Мухамедеева*
Технический редактор *Ф.А. Амирзянов*
Редактор *З.Г. Сафаралеева*

Отпечатано в Издательско-полиграфическом центре
Набережночелнинского института
Казанского (Приволжского) федерального университета

Подписано в печать 12.03.2015г.
Формат 60x84/16. Печать ризографическая
Бумага офсетная. Гарнитура «Times New Roman»
Усл. п. л. 7,5 Уч.-изд. л. 7,5
Тираж 300 экз. Заказ № 431 – 58