



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ
РЕСПУБЛИКИ БЕЛАРУСЬ**

**Белорусский национальный
технический университет**

Кафедра «Строительная механика»

**ЧИСЛЕННЫЕ МЕТОДЫ
РЕШЕНИЯ ЗАДАЧ СТРОИТЕЛЬСТВА**

Лабораторный практикум

**Минск
БНТУ
2016**

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Белорусский национальный технический университет

Кафедра «Строительная механика»

ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧ СТРОИТЕЛЬСТВА

Лабораторный практикум
для студентов специальности 1-70 02 01
«Промышленное и гражданское строительство»

*Рекомендовано учебно-методическим объединением
высших учебных заведений по образованию в области строительства
и архитектуры в качестве лабораторного практикума*

Минск
БНТУ
2016

УДК 69:519.6
ББК 22.193:38.112
Ч67

Составители:

А. В. Стрелюхин, Г. С. Богомолова, Е. Л. Сорокина

Рецензенты:

Р. И. Фурунжиев, А. С. Кравчук

Ч67 **Численные** методы решения задач строительства: лабораторный практикум для студентов специальности 1-70 02 01 «Промышленное и гражданское строительство» / сост. : А. В. Стрелюхин, Г. С. Богомолова, Е. Л. Сорокина. – Минск : БНТУ, 2016. – 116 с.
ISBN 978-985-550-512-0.

Настоящий практикум составлен на основе рабочей программы по курсу «Численные методы решения задач строительства». Материал состоит из введения, восьми лабораторных работ по численным методам решения задач линейной алгебры, дифференциальных уравнений с начальными и краевыми условиями, задач оптимизации, основам работы в системе компьютерной алгебры «Mathematica», списка литературы и задач для самостоятельного решения.

УДК 69:519.6
ББК 22.193:38.112

ISBN 978-985-550-512-0

© Белорусский национальный
технический университет, 2016

СОДЕРЖАНИЕ

Введение	4
Лабораторная работа № 1. Введение в систему компьютерной алгебры Mathematica.....	5
Лабораторная работа № 2. Решение систем линейных алгебраических уравнений методом Гаусса	20
Лабораторная работа № 3. Решение систем линейных алгебраических уравнений итерационными методами	31
Лабораторная работа № 4. Нахождение наибольшего по модулю собственного значения и соответствующего ему собственного вектора квадратной матрицы.....	43
Лабораторная работа № 5. Решение задачи Коши методами Рунге-Кутты.....	57
Лабораторная работа № 6. Решение краевой задачи для линейных дифференциальных уравнений методом конечных разностей	73
Лабораторная работа № 7. Численные методы оптимизации. Графический метод решения задач линейного программирования.....	88
Лабораторная работа № 8. Симплекс-метод решения задач линейного программирования.....	99
Литература	112
Приложение.....	113

ВВЕДЕНИЕ

Лабораторный практикум предназначен для студентов дневной и заочной формы обучения специальности 1-70 02 01 «Промышленное и гражданское строительство».

Успешное освоение учебных дисциплин: «Строительная механика», «Теоретическая механика», «Теория упругости» и т. д., составляющих основу знаний современного инженера-строителя, – требует комплексной подготовки, в том числе и владения математическим аппаратом. Курс «Численные методы решения задач строительства» носит междисциплинарный базовый характер, позволяет применять полученные знания для расчета заданий по другим учебным дисциплинам и решать инженерные и экономические задачи строительства.

Практикум состоит из восьми лабораторных работ, тематика которых охватывает основные разделы курса «Численные методы решения задач строительства». Каждый раздел начинается с изложения теоретических основ методов (решение задач линейной алгебры, дифференциальных уравнений с начальными и краевыми условиями и оптимизации), далее рассматриваются типовые примеры с краткими указаниями, поясняющими алгоритм решения. Дополнительную информацию по разделам можно получить из литературы, список которой приведен в конце практикума.

В настоящее время одним из основных направлений развития программного обеспечения является разработка систем компьютерной алгебры, позволяющих переложить многие громоздкие и трудоемкие вычисления на компьютер и сосредоточиться на проблеме постановки задачи и анализе результатов ее решения. Среди таких систем широкое применение получила система компьютерной алгебры «*Mathematica*». В связи с этим в практикуме уделяется особое внимание изучению основ работы с системой «*Mathematica*» и применению полученных знаний для решения конкретных задач в соответствии с программой курса.

Так же в практикуме приводятся задачи для самостоятельного решения.

ЛАБОРАТОРНАЯ РАБОТА № 1

ВВЕДЕНИЕ В СИСТЕМУ КОМПЬЮТЕРНОЙ АЛГЕБРЫ MATHEMATICA

Цель работы: изучить основные возможности системы компьютерной алгебры *Mathematica* и уметь применять ее для инженерных расчетов.

Состав работы:

1. Краткое описание системы компьютерной алгебры *Mathematica*;
2. Работа со списками, векторами и матрицами;
3. Работа с графикой;
4. Функция пользователя;
5. Выполнение индивидуальных заданий.

1.1. Краткое описание системы компьютерной алгебры *Mathematica*

Решение многих инженерных задач требует проведения достаточно большого количества расчетов. Кроме того, очень часто приходится обрабатывать полученные результаты, а также представлять их в наглядном виде. Выполнение таких операций вручную является достаточно трудоемкой работой. В связи с этим был разработан целый ряд программных продуктов (так называемые системы компьютерной алгебры *Computer Algebra System (CAS)*) для обработки числовых, символьных и графических данных, которые интегрируют в себе современный интерфейс пользователя, возможности решения задач, как аналитически, так и численно, а также мощные средства визуализации полученных результатов с использованием различных типов графиков.

Примером такого программного продукта является система компьютерной алгебры *Mathematica*, разработанная фирмой *Wolfram Research Inc.* Первая версия системы появилась в 1988 г. В 2013 г. вышла девятая версия системы, а в 2014 г. – десятая версия.

Возможности системы: *аналитические преобразования* (решение систем полиномиальных и тригонометрических уравнений и неравенств, трансцендентных уравнений, нахождение пределов, интегрирование и дифференцирование функций, решение дифференци-

альных уравнений и уравнений в частных производных и т. д.); *численные расчеты* (решение систем уравнений, нахождение пределов, интегрирование и дифференцирование, решение дифференциальных уравнений и уравнений в частных производных и т. д.); *линейная алгебра* (операции с матрицами, поиск собственных значений и собственных векторов); *графика* и *звук* и многое другое.

Запуск системы на компьютере под управление операционной системы *Windows* (при стандартной установке) осуществляется следующим способом

Пуск – Программы – Wolfram Mathematica – Wolfram Mathematica 9

Интерфейс системы состоит из нескольких окон. На рисунке показан интерфейс программы с открытыми окнами главного меню системы, ввода информации для вычислений и окна **Basic Math Assistant**.

Кратко рассмотрим содержание отдельных пунктов главного меню:

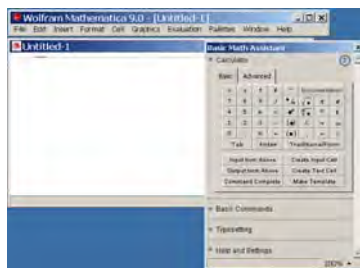
File – предназначен для создания нового окна документа, открытия существующего на внешнем носителе файла, закрытия окна документа, сохранения полученных результатов в файл, предварительного просмотра документа и его печати;

Edit – содержит стандартные команды для работы с буфером обмена, поиском информации в тексте документа и настройке системы;

Palettes – позволяет вызвать палитры (окна с набором пиктограмм), с помощью которых можно получить быстрый и удобный доступ к вводу математических символов и функций;

Window – содержит стандартные команды для работы с окнами;

Help – содержит команды для доступа к справочной информации, например к **Documentation Center** и **Function Navigator**, а также доступ к *Web* узлу компании *Wolfram*.



Работа с пунктами меню и их подпунктами осуществляется аналогично другим приложениям под управлением операционной системы Windows.

Ввод информации для расчетов основан на концепции рабочего поля или окна документа, называемого **блокнотом** (*notebook*). В нем могут находиться вычисления, текст, графика, звук и т. д., что в дальнейшем может использоваться для создания отчетов, презентаций и книг. Блокноту соответствует файл текстового формата с расширением *.nb*.

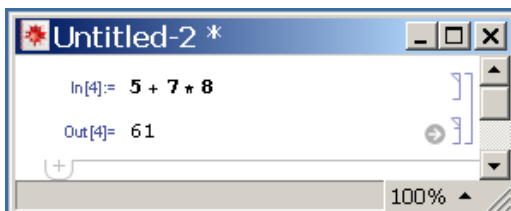
Основу блокнота составляют **ячейки** (*cells*). Ячейки указаны правыми скобками различных форм (по умолчанию скобка квадратная **]**) и могут содержать в себе другие ячейки.

Наиболее простой способ работы с системой *Mathematica* – интерактивный. В блокноте ввод данных (передача информации от «пользователя» к «вычислительному ядру» системы) производится нажатием клавиш

[Enter] [расширенной клавиатуры] или **[Shift-Enter]**.

Клавиша **[Enter]** [основной клавиатуры] дает возможность вводить несколько команд.

Например, если мы хотим вычислить выражение $5 + 7 \cdot 8$, необходимо набрать это выражение в окне ввода и запустить текущий блок



на выполнение нажатием клавиш **[Shift-Enter]**. В результате выполнения данной операции получим результат 61.

Признаком выполнения расчета является горизонтальная черта. В начале ввода информации на месте черты появится новый блок, отмеченный новой квадратной скобкой, в котором и будет отображаться вводимая информация.

При вычислениях исходное выражение присваивается объекту **In[n]**, а результат вычисления – объекту **Out[n]**, где **n** – номер проводимого вычисления. Значение **n** можно использовать в дальнейших вычислениях. Объекты **In[n]** и **Out[n]** пользователем не вводятся за исключением случая обращения к конкретному объекту **In[n]** или **Out[n]**.

Например, прибавим 2 к полученному в предыдущем примере результату.

```
In[4]:= 5 + 7 * 8
Out[4]= 61

In[5]:= Out[4] + 2
Out[5]= 63
```

Точка с запятой (;) в конце записанного выражения предотвращает вывод результата, но само вычисление выполняется.

Комментарий в блокноте записывается с использованием (* ... *)

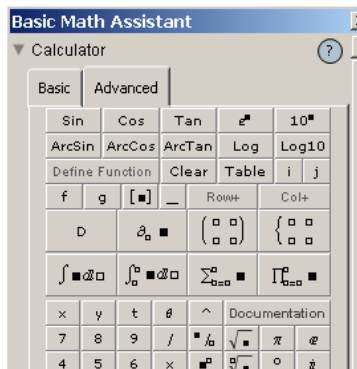
Символы %, %% и %n возвращают результат последней, предпоследней и в строке n операции.

Математические выражения в системе *Mathematica* записываются с помощью операторов и функций. Ниже перечислены основные операторы для выполнения арифметических действий

Операция	Знак	Операция	Знак
скобки	()	Деление	/
факториал	!	Умножение	* или [пробел]
двойной факториал	!!	Сложение	+
Возведение в степень	^	Вычитание	-

Намного упрощают работу по подготовке документов кнопочные палитры, предназначенные для ввода математических знаков. Если это окно не отображено, его можно вызвать из главного меню **Palettes - Basic Math Assistant**.

Вместе с тем, для большинства операций можно использовать при вводе соответствующий набор клавиш, например



Операция	Комбинация клавиш	Результат	Операция	Комбинация клавиш	Результат
Степень	x[Ctrl+6]5	x^5	Корень квадратный	[Ctrl+2]x	\sqrt{x}
Дробь	x[Ctrl+ /]5	$\frac{x}{5}$	Радиал	[Ctrl+2]x [Ctrl+5]5	$\sqrt[5]{x}$

Система *Mathematica* имеет достаточно большой набор предопределенных констант и функций, встроенных в ядро системы *Mathematica*, которые можно использовать при вычислениях.

Основные константы

Обозначение	Описание
Pi	константа, хранящая значение числа $\pi = 3,14159\dots$
E	константа, хранящая значение числа $e = 2,71828\dots$
I	мнимая единица
Infinity или ∞	знак бесконечность используется при некоторых вычислениях, например, при вычислении пределов, сумм и интегралов
Degree	константа для преобразования радиан в градусы, которая численно равно количеству радиан в одном градусе

Основные элементарные функции

Обозначение	Описание
Sqrt[x]	квадратный корень
Abs[x]	абсолютное значение действительного или модуль комплексного числа x
Mod[n,m]	остаток от деления n на m
Exp[x]	показательная функция e^x
Log[x]	натуральный логарифм $\ln(x)$
Log[a,x]	логарифм по основанию a ($\log_a x$)
Sign[x]	возвращает -1 , 0 или 1 , если аргумент x соответственно отрицательный, нулевой или положительный

Тригонометрические функции

Sin[x]	синус $\sin(x)$, x в радианах
Sin[x Degree]	синус $\sin(x)$, x в градусах
Cos[x]	косинус $\cos(x)$, x в радианах
Tan[x]	тангенс $\operatorname{tg}(x)$, x в радианах
Cot[x]	котангенс $\operatorname{ctg}(x)$, x в радианах

Обратные тригонометрические функции

ArcSin[x]	арксинус $\arcsin(x)$
ArcCos[x]	арккосинус $\arccos(x)$
ArcTan[x]	арктангенс $\arctg(x)$
ArcCot[x]	арккотангенс $\text{arcctg}(x)$

Гиперболические функции

Sinh[x]	гиперболический синус
Cosh[x]	гиперболический косинус
Tanh[x]	гиперболический тангенс
Coth[x]	гиперболический котангенс

Функции для работы с комплексными числами

Arg[z]	аргумент комплексного числа z
Im[z]	мнимая часть комплексного числа z
Re[z]	действительная часть комплексного числа z
Conjugate[z]	комплексно-сопряженное с z число
Sign[z]	отношение $z/ z $ комплексного числа z

Полный перечень встроенных функций можно найти в справочной системе программы (главное меню **Help - Function Navigator**).

Справку по конкретной функции можно получить из окна документа, установив на ней курсор и нажав клавишу **F1**, или непосредственно в ячейке, например

?Plot	вызов справки по указанной функции
??Plot	вызов расширенной справки по указанной функции
?Plo*	вызов всех функций, содержащих первые буквы Plo в своем имени
?*Plo*	вызов всех функций, содержащих буквы Plo в своем имени

При наборе выражений в системе *Mathematica* существуют определенные правила:

- дробная и вещественная часть числа отделяется точкой;
- запятая разделяет аргументы функции;
- заглавные и прописные буквы различаются;
- имена встроенных функций всегда начинаются с заглавной буквы, например **Cos**. Если имя функции составное, т. е. состоит из нескольких корней, то каждая составная часть в имени функции начинается с заглавной буквы, например **ArcCos**.

– использование скобок различных типов жестко регламентировано:

круглые скобки () используются для группировки выражений и изменения порядка вычисления выражений, т. е. как и при обычных математических вычислениях. Использование других типов скобок, как это принято при обычных математических вычислениях, не допускается.

квадратные скобки [] являются основным признаком функций и используются при записи функций, внутри которых записываются аргументы, разделенные запятыми, например **Cos[x]**.

фигурные скобки { } используются при работе с массивами, списками и матрицами.

Проиллюстрируем полученную информацию на примере вычисления выражения $10\sin(\pi/2)(1 - 5\cos(\pi))$.

```
In[20]:= 10 * Sin[Pi / 2] *  
         (1 - 5 * Cos[Pi])  
Out[20]= 60
```

Так как изначально система *Mathematica* разрабатывалась для проведения аналитических расчетов, то не все выражения по умолчанию вычисляются численно. Например:

```
In[22]:= Sin[5]  
Out[22]= Sin[5]
```

Для того, чтобы получить результат в численном виде, существуют два способа.

Первый способ обычно используется для простых выражений и заключается в представлении числа как числа с плавающей запятой.

Вторым способом является использование специальной функции, которая сообщает системе, что данное выражение необходимо вычислить численно.

Эта функция называется **N** и имеет формат записи **N[выражение, точность]**. Если точность не указывается, то вычисления проводятся по умолчанию в пять знаков после точки. Результат использования различных способов показан на рисунке.

```
In[23]:= Sin[5.1]
Out[23]= -0.958924

In[24]:= N[Sin[5], 7]
Out[24]= -0.9589243

In[25]:= Sin[5];
          N[%]
Out[26]= -0.958924
```

При вычислении сложных выражений часто оказывается неудобно записывать его полностью. Тогда при его составлении можно воспользоваться переменными, которым присваивают результат промежуточного вычисления или некоторое значение. Имя переменной всегда начинается с буквы. Удобством переменных является то, что ими можно воспользоваться в любой момент после того как она была инициализирована и в любом месте вычислений. Следует помнить, что значения переменных, после того как они были просчитаны, хранятся только в течение текущего сеанса вычислений. При следующем запуске системы их значения автоматически не восстанавливаются – требуется заново запустить соответствующие ячейки на расчет.

Пример записи составного выражения показан на рисунке (в нем от вывода промежуточных результатов мы отказались с помощью символа **;**).

```
In[36]:= a = 1.;
          b = 2.;
          c = 10.;
          a + b / c
Out[36]= 1.2
```

Рассмотреть все возможности системы *Mathematica* не представляется возможным, поэтому остановимся только на основных вычислениях. Дополнительные возможности системы будут рассмотрены в последующих лабораторных работах.

1.2. Работа со списками, векторами и матрицами

Списки являются не только эффективным средством работы с выражениями в процессе численных и символьных вычислений в системе *Mathematica*, но и одним из наиболее фундаментальных способов структурирования данных множественного типа в виде массивов, матриц и др. Основные сведения о матрицах и действиях над ними находятся в Приложении.

В системе *Mathematica* списки задаются с использованием фигурных скобок $\{\}$. Например:

$\{1, 2, 3\}$	список из 3 целых чисел
$\{1.5, 2.3, 3.7\}$	список из 3 вещественных чисел
$\{a, b, c\}$	список из 3 символьных данных
$\{\{a, b\}, \{c, d\}\}$	список, эквивалентный матрице $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$

С помощью списков можно задавать более привычные типы сложных данных, например, векторы – одномерные массивы данных, матрицы – двумерные массивы и многомерные массивы. При этом каждая группа элементов многомерных списков выделяется своей парой фигурных скобок, а в целом список выделяется общими скобками.

Например, список $\{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}\}$ представляет собой матрицу 3 порядка

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}.$$

Списки можно составлять напрямую, задавая объекты в соответствии с описанным синтаксисом. Списки могут быть объектами присваивания значений переменным, например: $v := \{1, 2, 3, 4, 5\}$

Списки характеризуются размером, который представляет собой произведение числа элементов списков по каждому направлению. Число направлений называют размерностью. Например, одномерный список является вектором и характеризуется числом элементов по единственному направлению. При этом вектор может быть вектором-строкой и вектором-столбцом. Двумерный список представляет матрицу, имеющую m строк и n столбцов. Ее размер $m \times n$. Если $m = n$, матрица называется квадратной.

Особенно часто для генерации списков с элементами целого и вещественного типа используется функция **Table**, создающая таблицу-список.

Table[expr, {i, imin, imax}]

генерирует список значений **expr** при изменении **i** от значения **imin** до **imax** с шагом **1**

Table[expr, {i, imin, imax, istep}]

генерирует список значений **expr** при изменении **i** от значения **imin** до **imax** с шагом **istep**

Например:

<pre>In[125]= Table[3*x, {x, 0., 3.}] Out[125]= {0., 3., 6., 9.}</pre>	<pre>In[126]= Table[3*x, {x, 0., 3., 0.5}] Out[126]= {0., 1.5, 3., 4.5, 6., 7.5, 9.}</pre>
<pre>In[120]= f = Sin[x] Table[f, {x, 0., 2*Pi, Pi/4}] Out[120]= Sin[x] Out[130]= {0., 0.707107, 1., 0.707107, 1.22465*10^-16, -0.707107, -1., -0.707107, -2.44929*10^-16}</pre>	<pre>In[131]= Table[{x, 5*x}, {x, 0., 5., 0.5}] Out[131]= {{0., 0.}, {0.5, 2.5}, {1., 5.}, {1.5, 7.5}, {2., 10.}, {2.5, 12.5}, {3., 15.}, {3.5, 17.5}, {4., 20.}, {4.5, 22.5}, {5., 25.}}</pre>

Для вывода элементов списка используются следующие функции:

- **TableForm[список]** выполняет вывод с элементами списка **список**, расположенными в массиве прямоугольных элементов;
- **MatrixForm[список]** выводит **список** в форме матрицы.

<pre>In[138]= a = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}; TableForm[a] Out[139]//TableForm= 1 2 3 4 5 6 7 8 9</pre>	<pre>In[144]= TableForm[Table[{x, 5*x}, {x, 0., 5., 0.5}]] Out[144]//TableForm= 0. 0. 0.5 2.5 1. 5. 1.5 7.5 2. 10. 2.5 12.5 3. 15. 3.5 17.5 4. 20. 4.5 22.5 5. 25.</pre>
<pre>In[140]= a = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}; MatrixForm[a] Out[141]//MatrixForm= (1 2 3 4 5 6 7 8 9)</pre>	

Вектора в системе *Mathematica* трактуются как линейные, т. е. одноуровневые списки; **матрицы** – как двухуровневые. Другими словами можно сказать, что матрица является списком списков.

Рассмотрим основные функции, предназначенные для работы с векторами и матрицами.

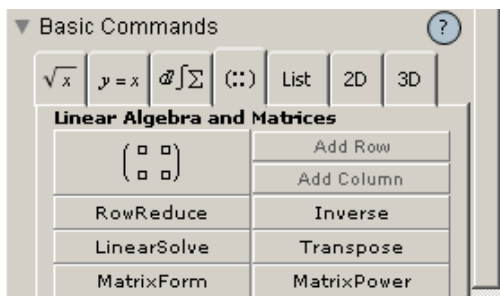
Обозначение	Описание
Det	вычисляет детерминант квадратной матрицы
Inverse	вычисляет обратную матрицу для невырожденных квадратных матриц
Transpose	вычисляет транспонированную матрицу
Dot[s1,s2] или s1.s2	вычисляет скалярное произведение векторов, произведение вектора и матрицы, а также произведение матриц
s1*s2	вычисляет поэлементное произведение списков
Cross[s1,s2]	для векторов вычисляет векторное произведение
+, -	сумма и разность векторов и матриц
Norm[s1]	вычисляет норму списка

Использование отдельных функций при работе с матрицами рассмотрим на следующих примерах.

Определитель матрицы Det	Умножение матрицы на вектор и матрицы на скаляр
<pre>In[92]:= m5 = {{10, -2, 4}, {-4, 15, 67}, {-7, 8, 19}}; MatrixForm[m5] Det[m5] Out[93]/MatrixForm= $\begin{pmatrix} 10 & -2 & 4 \\ -4 & 15 & 67 \\ -7 & 8 & 19 \end{pmatrix}$ Out[94]= -1432</pre>	<pre>In[23]:= m1 = {{a1, b1, c1}, {d1, e1, f1}, {g1, h1, i1}}; u = {p1, q1, s1}; m1.u; MatrixForm[%] t = 5; m1*t; MatrixForm[%] Out[28]/MatrixForm= $\begin{pmatrix} a1 p1 + b1 q1 + c1 s1 \\ d1 p1 + e1 q1 + f1 s1 \\ g1 p1 + h1 q1 + i1 s1 \end{pmatrix}$ Out[29]/MatrixForm= $\begin{pmatrix} 5 a1 & 5 b1 & 5 c1 \\ 5 d1 & 5 e1 & 5 f1 \\ 5 g1 & 5 h1 & 5 i1 \end{pmatrix}$</pre>

Примечание:

1. Для ввода элементов матрицы можно использовать соответствующую пиктограмму из палитры **Basic Math Assistant**. Для добавления нужного количества строк и столбцов используются кнопки **Add Row** и **Add Column**.



2. Для выполнения поэлементных действий с матрицами используются двойные квадратные скобки: $\mathbf{m1}[[1,1]]+\mathbf{m2}[[2,3]]$.

1.3. Работа с графикой

Система Mathematica имеет в своем распоряжении не очень большой набор функций для построения графиков, но, несмотря на это, она позволяет строить практически все типы графиков. Благодаря большому количеству опций и директив для графических функций система позволяет строить достаточно продвинутые и сложные графики с красивым оформлением.

1.3.1. Построение графиков функций, заданных аналитически

Все функции для построения двухмерных графиков оканчиваются на **Plot**, а трехмерных – на **Plot3D**.

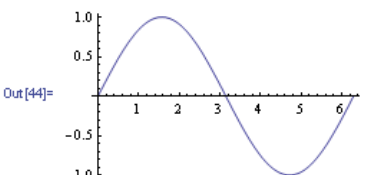
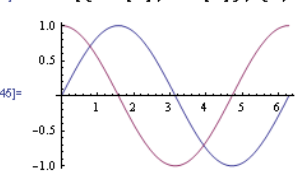
Plot[f(x), {x, xmin, xmax}].

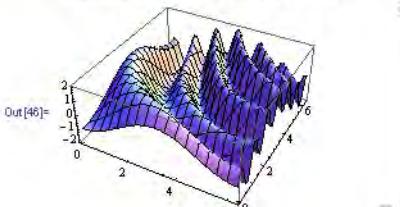
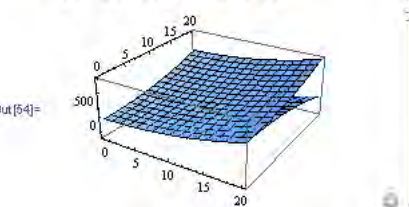
Построение графика функции $f(x)$ в декартовых координатах для x в пределах от x_{\min} до x_{\max} .

Plot3D[f(x,y), {x, xmin, xmax}, {y, ymin, ymax}].

Построение трехмерного графика функции $f(x, y)$ в декартовых координатах при изменении аргументов функции от минимального и до максимального значения.

Использование функций для построения графиков рассмотрим на следующих примерах.

Функция одной переменной	Две функции одной переменной на одном графике
<p>In[44]= <code>Plot[Sin[x], {x, 0, 2 * Pi}]</code></p>  <p>Out[44]=</p>	<p>In[46]= <code>Plot[{Sin[x], Cos[x]}, {x, 0, 2 * Pi}]</code></p>  <p>Out[46]=</p>

Функция двух переменных	Две функции двух переменных на одном графике
<p>In[46]= <code>Plot3D[Sin[x * y] - Cos[x + y], {x, 0, 2 * Pi}, {y, 0, 2 * Pi}]</code></p>  <p>Out[46]=</p>	<p>In[54]= <code>Plot3D[{x^2 + y^2, x^2 - y^2}, {x, 0, 20}, {y, 0, 20}]</code></p>  <p>Out[54]=</p>

Кроме того, в системе есть отдельные функции для построения графика функции, заданной в параметрической форме (**ParametricPlot**, **ParametricPlot3D**), контурного графика (**ContourPlot**), графика плотности (**DensityPlot**) и т. д. Более подробную информацию о них, а также об опциях для различного отображения графиков, можно получить из справочной системы программы.

1.3.2. Построение графиков функций, заданных таблично

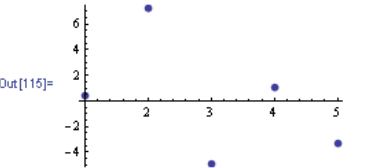
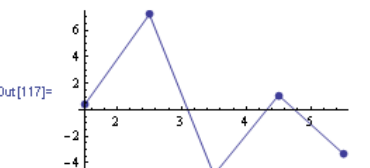
Далеко не все задачи можно решить в аналитическом виде. В этом случае на помощь приходят численные методы. Результатом таких вычислений являются табличные данные, которые хотелось бы отобразить в виде графиков. Система *Mathematica* имеет несколько встроенных функций, аналогичных рассмотренным ранее функциям.

Основные функции для построения графиков по точкам следующие:

- **ListPlot[список]**. Если в качестве параметра передается обычный список, то его значения будут взяты в качестве координат по y при построении графика, а координаты по x будут заданы из списка 1, 2, 3, Если же в качестве параметра будет передана матрица, состоящая из двух строк, то данные из первого столбца будут взяты в качестве x -координат, а второго столбца, соответственно, в качестве y -координат для точек. График отображается в виде точек.

- **ListLinePlot[список]**. Аналогична функции **ListPlot**, но график отображается в виде линий

- **ListPlot3D[матрица]**. По данным, представленным в матрице с тремя столбцами, строится поверхность, где столбцы задают значения x , y и z -координат.

График функции, заданный таблично	График функции, заданный таблично
<pre data-bbox="140 759 509 837">In[115]= ListPlot[{0.4, 7.2, -4.9, 1.1, -3.3}, PlotStyle -> PointSize[.03]]</pre>  <pre data-bbox="140 917 196 933">Out[115]=</pre>	<pre data-bbox="565 759 935 917">In[117]= ListLinePlot [{{1.5, 0.4}, {2.5, 7.2}, {3.5, -4.9}, {4.5, 1.1}, {5.5, -3.3}}, PlotStyle -> PointSize[.03], Mesh -> All]</pre>  <pre data-bbox="565 1013 621 1029">Out[117]=</pre>
<p>Координаты x заданы из списка 1, 2, 3, Размер точки установлен опцией PlotStyle</p>	<p>Координаты x и y заданы как элементы матрицы. Размер точки установлен опцией PlotStyle. Опция Mesh -> All отображает точки</p>

1.4. Функция пользователя

Хотя в системе *Mathematica* около тысячи встроенных функций, часто пользователю может потребоваться создание своей собст-

венной функции. Процесс создания функции пользователя рассмотрим на следующем примере.

```

▼ In[8]:= Myfun[x_] :=  $\frac{x + 10}{\sqrt{x^2 + 20}} + 5$ 
           {Myfun[5.], Myfun[w]}
Out[9]= {11.1803,  $\frac{5(10 + w)}{\sqrt{20 + w^2}}$ }

```

Здесь мы определили собственную функцию $y(x) = \frac{x + 10}{\sqrt{x^2 + 20}} \cdot 5$, присвоив ей имя **Myfun**. В ней x является формальным параметром и обязательно должен содержать символ подчеркивания при обращении к нему.

1.5. Выполнение индивидуальных заданий

Индивидуальное задание этой лабораторной работы подразумевает самостоятельное выполнение в системе *Mathematica* приведенных выше примеров.

ЛАБОРАТОРНАЯ РАБОТА № 2. РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ МЕТОДОМ ГАУССА

Цель работы: изучить метод Гаусса решения систем алгебраических уравнений.

Состав работы:

1. Изучение метода Гаусса (метод исключения неизвестных);
2. Изучение метода Гаусса с частичным выбором ведущего элемента;
3. Пример решения системы линейных алгебраических уравнений;
4. Решение систем линейных алгебраических уравнений в системе *Mathematica*;
5. Выполнение индивидуальных заданий.

2.1. Метод Гаусса (метод исключения неизвестных)

Пусть задана система линейных алгебраических уравнений (СЛАУ) n -го порядка

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1; \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2; \\ \dots\dots\dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n, \end{aligned} \tag{2.1}$$

где $n \geq 2$.

В матричном виде исходную систему можно представить как $\mathbf{A} \cdot \mathbf{X} = \mathbf{B}$ (основные сведения о матрицах находятся в Приложении).

Требуется найти решение системы (2.1), т. е. найти такие значения неизвестных x_1, x_2, \dots, x_n , которые при подстановке во все уравнения исходной системы обращают их в тождества. Из линейной алгебры известно, что система имеет единственное решение, если ее определитель не равен нулю, т. е. система является невырожденной.

Метод Гаусса состоит из двух основных этапов (ходов):

- 1) прямого хода;
- 2) обратного хода.

Цель прямого хода – преобразовать заданную систему к верхнему треугольному виду. Это значит, что первое уравнение новой системы сохранит свой прежний вид. Из второго уравнения исключается неизвестное x_1 , из третьего – x_1 и x_2 и т. д. Таким образом, последнее уравнение будет содержать только одно неизвестное x_n .

Цель обратного хода – непосредственно определить значения неизвестных, причем вначале определяется x_n , затем x_{n-1}, \dots, x_1 .

Рассмотрим прямой ход. Чтобы исключить x_1 из второго уравнения, найдем множитель $m = a_{21} / a_{11}$ (предполагаем, что коэффициент a_{11} , который в методе Гаусса называется первым **ведущим элементом**, отличен от нуля). Затем первое уравнение системы умножим на полученный множитель и почленно вычтем из второго уравнения. При этом его коэффициенты и свободный член получат новые значения

$$a_{21}^{(1)} = a_{21} - a_{11}m = a_{21} - a_{11} \frac{a_{21}}{a_{11}} = 0;$$

$$a_{22}^{(1)} = a_{22} - a_{12}m;$$

.....

$$a_{2n}^{(1)} = a_{2n} - a_{1n}m;$$

$$b_2^{(1)} = b_2 - b_1m.$$

В результате мы получили, что во втором уравнении отсутствует x_1 .

Чтобы исключить x_1 из третьего и последующих уравнений, надо найти множители $m = a_{31}/a_{11}, \dots, m = a_{n1}/a_{11}$, умножить на них все то же первое уравнение и почленно вычесть его из третьего и последующих уравнений.

В результате всех этих действий заданная система примет вид

$$\underline{a_{11}}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1;$$

$$\underline{a_{22}^{(1)}}x_2 + \dots + a_{2n}^{(1)}x_n = b_2^{(1)};$$

.....

$$a_{n2}^{(1)}x_2 + \dots + \underline{a_{nn}^{(1)}}x_n = b_n^{(1)},$$

где первый и второй **ведущие элементы** специально выделены.

Переходим к исключению x_2 из третьего и последующих уравнений, полагая, что второй ведущий элемент $a_{22}^{(1)}$ не равен нулю. Находя множители $m = a_{32}^{(1)} / a_{22}^{(1)}$, ..., $m = a_{n2}^{(1)} / a_{22}^{(1)}$, умножаем на них (поочередно) второе уравнение и почленно вычитаем из третьего, ..., n -го уравнений.

Аналогично проводится исключение x_3, \dots, x_{n-1} .

Все действия, направленные на исключение неизвестного x_i из j -го уравнения, можно представить тремя формулами:

$$\begin{aligned} m &= a_{ji}^{(i-1)} / a_{ii}^{(i-1)}; \\ a_{jl}^{(i)} &= a_{jl}^{(i-1)} - a_{il}^{(i-1)} m \quad (l = i+1, \dots, n); \\ b_j^{(i)} &= b_j^{(i-1)} - b_i^{(i-1)} m. \end{aligned} \quad (2.2)$$

Проведя вычисления по формулам (2.2) при изменении i от 1 до $n-1$, а j от $i+1$ до n , получим систему верхнего треугольного вида

$$\begin{aligned} \underline{a_{11}} x_1 + a_{12} x_2 + \dots + a_{1n} x_n &= b_1; \\ \underline{a_{22}^{(1)}} x_2 + \dots + a_{2n}^{(1)} x_n &= b_2^{(1)}; \\ &\dots\dots\dots \\ \underline{a_{nn}^{(n-1)}} x_n &= b_n^{(n-1)}, \end{aligned}$$

в которой ни один из **ведущих элементов** не должен быть равен нулю.

Рассмотрим обратный ход. В последнее уравнение входит только одно неизвестное x_n . Поэтому оно определится как

$$x_n = b_n^{(n-1)} / a_{nn}^{(n-1)}. \quad (2.3)$$

Предпоследнее уравнение преобразованной системы имеет вид

$$a_{n-1,n-1}^{(n-2)} x_{n-1} + a_{n-1,n}^{(n-2)} x_n = b_{n-1}^{(n-2)}.$$

Подставляя в него найденное значение x_n , получим

$$x_{n-1} = \left(b_{n-1}^{(n-2)} - a_{n-1,n}^{(n-2)} x_n \right) / a_{n-1,n-1}^{(n-2)}.$$

Продолжая этот процесс, достигнем первого уравнения, из которого

$$x_1 = (b_1 - a_{12}x_2 - \dots - a_{1n}x_n) / a_{11}.$$

Все действия, совершаемые на обратном ходе при определении $x_{n-1}, x_{n-2}, \dots, x_1$, можно описать одной формулой

$$x_i = \left(b_i^{(i-1)} - \sum_{s=i+1}^n a_{is}^{(i-1)} x_s \right) / a_{ii}^{(i-1)}, \quad (2.4)$$

где $i = n-1, \dots, 1$.

После окончания обратного хода найденные значения x_1, x_2, \dots, x_n необходимо подставить во все уравнения исходной системы, чтобы убедиться в том, что они обращаются в тождества и что x_1, x_2, \dots, x_n образуют решение.

Примечания.

1. Решая систему методом Гаусса, надо контролировать, чтобы какой-нибудь ведущий элемент не оказался равным нулю, так как деление на нуль невозможно, а именно в операции деления и участвуют ведущие элементы. Нулевой ведущий элемент может появиться в двух случаях. Во-первых, когда система вырожденная и ее определитель равен нулю. В этом случае решение вообще не существует либо имеется множество решений. Во-вторых, заданная система, будучи невырожденной, может иметь коэффициент $a_{11} = 0$. Для этого используют, например, метод Гаусса с выбором ведущего элемента (раздел 2.2).

2. Описанный формулами (2.2) – (2.4) алгоритм можно обобщить на случай решения системы с k правыми частями. В этом случае свободные члены и неизвестные имеют два индекса и образуют матрицы (массивы) с размерами $n \times k$. Третья из формул (2.2), а также формулы (2.3) и (2.4) теперь будут выглядеть так

$$\begin{aligned}
 b_{jl}^{(i)} &= b_{jl}^{(i-1)} - b_{il}^{(i-1)} m; \\
 x_{nl} &= b_{nl}^{(n-1)} / a_{nn}^{(n-1)}; \\
 x_{il} &= (b_{il}^{(i-1)} - \sum_{s=i+1}^n a_{is}^{(i-1)} x_{sl}) / a_{ii}^{(i-1)},
 \end{aligned}$$

где $l = 1, \dots, k$.

3. Метод Гаусса решения СЛАУ можно применить для вычисления определителя матрицы \mathbf{A} n -го порядка. В отличие от классической формулы, это выполняется очень просто, как произведение n ведущих элементов:

$$\Delta = (-1)^\gamma a_{11} a_{22}^{(1)} \dots a_{nn}^{(n-1)} = (-1)^\gamma \cdot \prod_{i=1}^n a_{ii}^{(i-1)}, \quad (2.5)$$

где γ – число перестановок строк при выборе ведущего элемента в процессе преобразования исходной матрицы \mathbf{A} к треугольному виду.

4. Метод Гаусса можно применить для поиска обратной матрицы.

Исходя из определения обратной матрицы $\mathbf{A} \cdot \mathbf{A}^{-1} = \mathbf{E}$ и используя для наглядности обозначение $\mathbf{X} = \mathbf{A}^{-1}$, приходим к системе линейных алгебраических уравнений вида $\mathbf{A} \cdot \mathbf{X} = \mathbf{E}$, решение которой и будет составлять элементы обратной матрицы.

2.2. Метод Гаусса с частичным выбором ведущего элемента

Пусть требуется решить систему трех уравнений

$$\begin{aligned}
 -4,2x_2 + 8,1x_3 &= 40,5, \\
 2,2x_1 + 4,1x_2 + 6,3x_3 &= -5,6, \\
 6,8x_1 + 2,2x_2 - 3,4x_3 &= 14,3.
 \end{aligned}$$

Данная система является невырожденной (определитель равен – 397,07), следовательно имеет единственное решение. Но в данном случае метод Гаусса неприменим, т.к. первый ведущий элемент $a_{11} = 0$.

У метода Гаусса есть еще один существенный недостаток: ведущий элемент может оказаться не точным нулем, а очень малой величиной. Тогда соответствующие множители получатся очень большими. Они внесут в преобразованные коэффициенты и свободные члены большие погрешности, способные накапливаться в процессе дальнейшего решения. В таком случае полученные результаты могут оказаться неудовлетворительными.

Выход из этих ситуаций заключается в выборе ведущего элемента. Самый простой вариант такого выбора состоит в том, что перед тем, как исключить x_1 из второго и последующих уравнений, сравниваются между собой коэффициенты первого столбца $a_{11}, a_{21}, \dots, a_{n1}$, среди которых выбирается наибольший по модулю. Пусть таковым будет не a_{11} , а некоторый другой – a_{k1} . Тогда, поменяв местами первое и k -е уравнения, на месте a_{11} получим величину, наибольшую из всех коэффициентов первого столбца. Вследствие этого соответствующие множители будут по модулю не большими, чем единица.

В примере наибольший коэффициент первого столбца находится в третьем уравнении ($k = 3$). Поэтому меняем местами первое и третье уравнения:

$$\begin{aligned} 6,8x_1 + 2,2x_2 - 3,4x_3 &= 14,3, \\ 2,2x_1 + 4,1x_2 + 6,3x_3 &= -5,6, \\ -4,2x_2 + 8,1x_3 &= 40,5. \end{aligned}$$

Определив множители $m = 2,2/6,8 \approx 0,324$ и $m = 0/6,8 = 0$, преобразуем эту систему к виду

$$\begin{aligned} 6,8x_1 + 2,2x_2 - 3,4x_3 &= 14,3 \\ -3,388x_2 - 7,4x_3 &= 10,227 \\ -4,2x_2 + 8,1x_3 &= 40,5. \end{aligned}$$

Перед исключением x_2 из третьего и последующих уравнений выбор наибольшего по модулю ведется среди коэффициентов $a_{22}^{(1)}, \dots, a_{n2}^{(1)}$. В рассматриваемом примере наибольшим по модулю будет $a_{32}^{(1)} = -4,2$. Меняем местами второе и третье уравнения

$$\begin{array}{rclcrcl}
6,8x_1 & +2,2x_2 & -3,4x_3 & = & 14,3, \\
& -4,2x_2 & +8,1x_3 & = & 40,5, \\
-3,388x_2 & -7,4x_3 & & = & 10,227.
\end{array}$$

Определив множитель $m = -3,388/(-4,2) \approx 0,807$, приходим к системе

$$\begin{array}{rclcrcl}
6,8x_1 & +2,2x_2 & -3,42x_3 & = & 14,3, \\
& -4,2x_2 & +8,1x_3 & = & 40,5, \\
& & 13,937x_3 & = & 22,457.
\end{array}$$

Используя обратный ход, получаем $x_1 = 1,611$; $x_2 = -6,536$, $x_3 = 5,028$.

Определитель матрицы \mathbf{A} определим с помощью выражения (2.5)

$$\det|\mathbf{A}| = (-1)^2 \cdot 6,8 \cdot (-4,2) \cdot 13,937 \approx -398,04.$$

2.3. Пример решения системы линейных алгебраических уравнений

Решить СЛАУ третьего порядка методом Гаусса

$$\begin{array}{rclcrcl}
25,2x_1 & -4,7x_2 & +1,4x_3 & = & 46,8, \\
4,1x_1 & +39,4x_2 & +2,9x_3 & = & -25,6, \\
-1,3x_1 & +3,1x_2 & +16,8x_3 & = & 34,3
\end{array}$$

Результаты представить с четырьмя значащими цифрами.

Перед тем, как решать задачу, необходимо вычислить определитель системы: $\det|\mathbf{A}| = 16884,793$, т. е. матрица является невырожденной и решение этой системы единственно.

Выполним прямой ход.

Исключаем x_1 из второго уравнения. Для этого находим множитель $m = a_{21} / a_{11} = 4,1 / 25,2 = 0,162698$, первое уравнение системы умножаем на этот множитель и почленно вычитаем его из второго уравнения:

$$\begin{aligned}
 a_{22}^{(1)} &= a_{22} - a_{12}m = 39,4 - (-4,7) \cdot 0,162698 = 40,1647, \\
 a_{23}^{(1)} &= a_{23} - a_{13}m = 2,9 - 1,4 \cdot 0,162698 = 2,67222, \\
 b_2^{(1)} &= b_2 - b_1m = -25,6 - 46,8 \cdot 0,162698 = -33,2143.
 \end{aligned}$$

Исключаем x_1 из третьего уравнения аналогичным образом:

$$\begin{aligned}
 m &= a_{31} / a_{11} = -1,3 / 25,2 = -0,0515873, \\
 a_{32}^{(1)} &= a_{32} - a_{12}m = 3,1 - (-4,7) \cdot (-0,0515873) = 2,85754, \\
 a_{33}^{(1)} &= a_{33} - a_{13}m = 16,8 - 1,4 \cdot (-0,0515873) = 16,8722, \\
 b_3^{(1)} &= b_3 - b_1m = 34,3 - 46,8 \cdot (-0,0515873) = 36,7143.
 \end{aligned}$$

В результате получили преобразованную систему:

$$\begin{array}{rclcl}
 \underline{25,2}x_1 & -4,7x_2 & +1,4x_3 & = & 46,8, \\
 & \underline{40,1647}x_2 & +2,67222x_3 & = & -33,2143, \\
 & 2,85753x_2 & +16,8722x_3 & = & 36,7143.
 \end{array}$$

Исключаем x_2 из третьего уравнения:

$$\begin{aligned}
 m &= a_{32}^{(1)} / a_{22}^{(1)} = 2,85753 / 40,1647 = 0,0711453, \\
 a_{33}^{(2)} &= a_{33}^{(1)} - a_{23}^{(1)}m = 16,8722 - 2,67222 \cdot 0,0711453 = 16,6821, \\
 b_3^{(2)} &= b_3^{(1)} - b_2^{(1)}m = 36,7143 - (-33,2143) \cdot 0,0711453 = 39,0773.
 \end{aligned}$$

Окончательно преобразованная система имеет вид

$$\begin{array}{rclcl}
 \underline{25,2}x_1 & -4,7x_2 & +1,4x_3 & = & 46,8, \\
 & \underline{40,1647}x_2 & +2,67222x_3 & = & -33,2143, \\
 & & \underline{16,6821}x_3 & = & 39,0773.
 \end{array}$$

Теперь выполним обратный ход:

$$\begin{aligned}x_3 &= b_3^{(2)} / a_{33}^{(2)} = 39,0773 / 16,6821 = 2,34247, \\x_2 &= (b_2^{(1)} - a_{23}^{(1)} x_3) / a_{22}^{(1)} = \\&= (-33,2143 - 2,67222 \cdot 2,34247) / 40,1647 = -0,982801, \\x_1 &= (b_1 - a_{12} x_2 - a_{13} x_3) / a_{11} = \\&= (46,8 - (-4,7) \cdot (-0,982801) - 1,4 \cdot 2,34247) / 25,2 = 1,54370.\end{aligned}$$

Проверка полученного решения.

Подставим полученные значения x_1 , x_2 , x_3 в левую часть исходной системы:

$$\begin{aligned}25,2 \cdot 1,54370 - 4,7 \cdot (-0,982801) + 1,4 \cdot 2,34247 &= 46,7999, \\4,1 \cdot 1,54370 + 39,4 \cdot (-0,982801) + 2,9 \cdot 2,34247 &= -25,6000, \\-1,3 \cdot 1,54370 + 3,1 \cdot (-0,982801) + 16,8 \cdot 2,34247 &= 34,3000.\end{aligned}$$

Найденные числа с достаточно высокой точностью совпадают с правыми частями заданной системы. Это значит, что система решена правильно. Значения неизвестных округляем до четырех значащих цифр: $x_1 = 1,544$, $x_2 = -0,9828$, $x_3 = 2,343$.

Для того, чтобы вычислить определитель матрицы **A** по методу Гаусса, воспользуемся выражением (2.5):

$$\det|\mathbf{A}| = (-1)^0 \cdot 25,2 \cdot 40,1647 \cdot 16,6821 \approx 16\,884,79.$$

2.4. Решение систем линейных алгебраических уравнений в системе Mathematica

Для решения систем линейных уравнений в системе *Mathematica* используется функция **LinearSolve[A,B]**, которая возвращает вектор **X**, являющийся решением матричного уравнения **A · X = B**.

Рассмотрим пример из п. 2.3 и воспользуемся системой *Mathematica* для решения.

```

▼ In[23]:= A =  $\begin{pmatrix} 25.2 & -4.7 & 1.4 \\ 4.1 & 39.4 & 2.9 \\ -1.3 & 3.1 & 16.8 \end{pmatrix}$ ;

B =  $\begin{pmatrix} 46.8 \\ -25.6 \\ 34.3 \end{pmatrix}$ ;

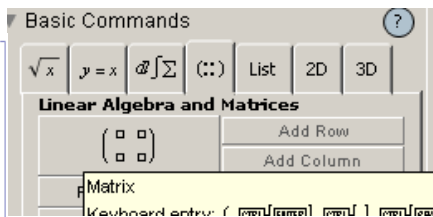
Det[A]
LinearSolve[A, B]
MatrixForm[%]

Out[25]= 16884.8

Out[26]= {{1.54371}, {-0.982801}, {2.34247}}

Out[27]/MatrixForm=
 $\begin{pmatrix} 1.54371 \\ -0.982801 \\ 2.34247 \end{pmatrix}$ 

```



Для ввода элементов матриц **A** и **B** использовалась пиктограмма **Matrix** из палитры **Basic Math Assistant** (вкладка **Matrix Commands**).

Нужное количества строк и столбцов устанавливается кнопками **Add Row** и **Add Column**

2.5. Выполнение индивидуальных заданий

Составить систему линейных алгебраических уравнений, выбрав коэффициенты и свободные члены из табл. 2.1 согласно шифру (см. указания к работе). Решить эту систему методом исключения Гаусса. Сделать проверку полученных результатов. Решить задание, используя систему Mathematica, и сравнить полученные результаты.

Таблица 2.1

Исходные данные к заданию

Первая цифра шифра	Коэффициенты системы уравнений			Вторая цифра шифра	Свободные члены b_i	Третья цифра шифра	Величина c
	a_{i1}	a_{i2}	a_{i3}				
1	2	3	4	5	6	7	8
0	$4 + c$	2,353	0,189	0	2,411	0	0,954
	1,207	3,125	1,424		-0,315		
	0,248	0,953	2,267		2,000		
1	$3,742$	$6 + c$	1,256	1	-1,940	1	-
	4,025	2,994	-0,984		1,688		
	1,562	0,738	4,092		2,198		

1	2	3	4	5	6	7	8
2	3,116 -2,158 2,352	2,125 3,734 1,616	3 + c 3,450 3,528	2	1,480 2,914 -2,165	2	0,305
3	2,691 2 + c 3,546	2,472 3,924 4,017	-2,073 1,925 3,638	3	-1,270 -2,424 1,372	3	- 0,167
4	3,886 3,004 -2,673	3,457 5 + c 2,392	1,968 3,785 4,101	4	1,801 -1,212 -2,624	4	0,425
5	5,116 2,158 3,195	2,125 4,168 -2,207	1,140 2 + c 4,218	5	-2,466 -1,215 1,272	5	- 0,203
6	4,071 2,848 3 + c	-3,797 2,442 4,504	3,376 1,955 3,973	6	2,837 -3,004 3,725	6	- 0,632
7	5,113 2,957 3,306	2,588 4,625 4 + c	-2,164 2,949 4,092	7	1,430 2,175 0,998	7	0,530
8	2,930 -3,468 2,504	-2,152 2,987 3,706	2,146 4,225 5 + c	8	-2,294 1,748 1,025	8	- 0,758
9	6 + c 4,436 2,407	2,890 4,029 -3,913	2,476 -2,923 4,738	9	2,500 -1,692 1,573	9	0,873

Указания к работе.

1. Исходные данные к контрольной работе выбираются студентом из таблицы в соответствии с его личным учебным шифром (номером зачетной книжки). Шифром считаются три последние цифры. Если номер зачетной книжки двузначный, то его следует записать дважды и взять три последние цифры.

2. Пусть из табл. 2.1 выбраны три группы величин:

Коэффициенты СЛАУ			Свободные члены	c
2,012	-1,347	0,872	1,438	-0,972
3,457	4 + c	-2,046	-2,562	
1,827	-3,767	6,329	0,899	

Имея их, надо сформировать СЛАУ

$$\begin{aligned}2,012x_1 - 1,347x_2 + 0,872x_3 &= 1,438, \\3,457x_1 + 3,028x_2 - 2,046x_3 &= -2,562, \\1,827x_1 - 3,767x_2 + 6,329x_3 &= 0,899.\end{aligned}$$

Здесь коэффициент $a_{22} = 3,028$ получен сложением 4 и c , равного $-0,972$.

ЛАБОРАТОРНАЯ РАБОТА № 3 РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ ИТЕРАЦИОННЫМИ МЕТОДАМИ

Цель работы: изучить итерационные методы решения систем линейных алгебраических уравнений – метод простой итерации и метод Зейделя.

Состав работы:

1. Изучение алгоритмов итерационных методов решения систем линейных алгебраических уравнений;
2. Пример решения системы линейных уравнений итерационными методами;
3. Выполнение индивидуальных заданий.

3.1. Итерационные методы решения систем линейных алгебраических уравнений

Рассмотрим систему линейных алгебраических уравнений n -го порядка

$$\begin{aligned}a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\&\dots\dots\dots \\a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n.\end{aligned}\tag{3.1}$$

Метод Гаусса предъявляет к решаемой системе единственное требование: она должна быть невырожденной. Для итерационных методов этого требования недостаточно – требуется дополнительное условие, которое называется *условием сходимости*. Причем имеются два таких условия: 1) необходимое и достаточное; 2) только достаточное. Однако проверка первого условия часто оказывается более трудоемкой операцией, чем само решение системы.

Мы будем пользоваться самым простым, достаточным условием сходимости, одинаковым для обоих методов. Оно называется *условием диагонального преобладания* и формулируется следующим образом: для сходимости итерационного процесса достаточно, чтобы модуль диагонального коэффициента был больше суммы модулей остальных коэффициентов данного уравнения

$$\begin{aligned} |a_{11}| &> |a_{12}| + \dots + |a_{1n}|, \\ |a_{22}| &> |a_{21}| + |a_{23}| + \dots + |a_{2n}|, \\ &\dots\dots\dots \\ |a_{nn}| &> |a_{n1}| + \dots + |a_{n,n-1}|. \end{aligned} \tag{3.2}$$

В отдельных случаях допускается строгое равенство.

В математике имеется строгое доказательство того факта, что если диагональные коэффициенты преобладают в каждом уравнении системы, то ее определитель отличен от нуля. Обратное утверждение, вообще говоря, неверно.

Итак, *первым* этапом в решении системы уравнений (3.1) итерационными методами является проверка условия диагонального преобладания (3.2).

Если оно выполняется, переходят ко *второму* этапу – преобразованию системы к виду, удобному для проведения итераций. Такое преобразование может быть выполнено бесконечно большим числом способов. Наиболее просто это делается путем разрешения первого уравнения относительно x_1 , второго – относительно x_2 и т. д.

Получаем:

$$x_1 = \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}} x_2 - \dots - \frac{a_{1n}}{a_{11}} x_n,$$

$$x_2 = \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}}x_1 - \frac{a_{23}}{a_{22}}x_3 \dots - \frac{a_{2n}}{a_{22}}x_n,$$

.....

$$x_n = \frac{b_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}}x_1 - \dots - \frac{a_{n,n-1}}{a_{nn}}x_{n-1}$$

Или

$$\begin{aligned} x_1 &= d_1 - c_{12}x_2 - \dots - c_{1n}x_n, \\ x_2 &= d_2 - c_{21}x_1 - c_{23}x_3 - \dots - c_{2n}x_n, \\ &\dots\dots\dots \\ x_n &= d_n - c_{n1}x_1 - \dots - c_{n,n-1}x_{n-1}, \end{aligned} \tag{3.3}$$

Где

$$d_i = b_i / a_{ii}, c_{ij} = a_{ij} / a_{ii} \quad (j \neq i), \quad c_{ii} = 0. \tag{3.4}$$

Система уравнений, приведенная к виду (3.3), называется **нормальной**.

На *третьем* этапе нужно задать начальные значения неизвестных $x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}$, которые называются **начальным приближением**.

Наиболее часто используются два варианта задания начальных значений:

1) $x_i^{(0)} = 0$; 2) $x_i^{(0)} = d_i$, где $i = 1, \dots, n$. Однако, если выполняется условие сходимости, итерационный процесс сходится к одним и тем же значениям неизвестных при задании в качестве начального приближения любых чисел.

На *четвертом* этапе строится итерационный процесс, состоящий в том, чтобы от любых чисел $x_i^{(0)}$ последовательными однотипными шагами (итерациями) приблизиться к истинным значениям x_i . Причем на первой итерации ($k = 1$), используя начальные значения, находят первое приближение $x_i^{(1)}$, на второй итерации ($k = 2$) – второе приближение $x_i^{(2)}$ и т. д.

Для вычисления $x_i^{(1)}$ в **методе простой итерации** начальные значения подставляют в правые части всех уравнений нормальной системы (3.3):

$$\begin{aligned}x_1^{(1)} &= d_1 - c_{12}x_2^{(0)} - \dots - c_{1n}x_n^{(0)}, \\x_2^{(1)} &= d_2 - c_{21}x_1^{(0)} - c_{23}x_3^{(0)} - \dots - c_{2n}x_n^{(0)}, \\&\dots\dots\dots \\x_n^{(1)} &= d_n - c_{n1}x_1^{(0)} - \dots - c_{n,n-1}x_{n-1}^{(0)}.\end{aligned}$$

Соответственно, для получения второго приближения $x_i^{(2)}$ в правые части нормальной системы надо подставить первое приближение:

$$\begin{aligned}x_1^{(2)} &= d_1 - c_{12}x_2^{(1)} - \dots - c_{1n}x_n^{(1)}, \\x_2^{(2)} &= d_2 - c_{21}x_1^{(1)} - c_{23}x_3^{(1)} - \dots - c_{2n}x_n^{(1)}, \\&\dots\dots\dots \\x_n^{(2)} &= d_n - c_{n1}x_1^{(1)} - \dots - c_{n,n-1}x_{n-1}^{(1)}.\end{aligned}$$

На k -й итерации ($k \geq 1$) находится k -е приближение $x_i^{(k)}$:

$$\begin{aligned}x_1^{(k)} &= d_1 - c_{12}x_2^{(k-1)} - \dots - c_{1n}x_n^{(k-1)}, \\x_2^{(k)} &= d_2 - c_{21}x_1^{(k-1)} - c_{23}x_3^{(k-1)} - \dots - c_{2n}x_n^{(k-1)}, \\&\dots\dots\dots \\x_n^{(k)} &= d_n - c_{n1}x_1^{(k-1)} - \dots - c_{n,n-1}x_{n-1}^{(k-1)}.\end{aligned} \tag{3.5}$$

Метод Зейделя является модификацией метода простой итерации и отличается от последнего только способом вычисления приближений:

$$\begin{aligned}x_1^{(k)} &= d_1 - c_{12}x_2^{(k-1)} - \dots - c_{1n}x_n^{(k-1)}, \\x_2^{(k)} &= d_2 - c_{21}x_1^{(k)} - c_{23}x_3^{(k-1)} - \dots - c_{2n}x_n^{(k-1)}, \\&\dots\dots\dots \\x_n^{(k)} &= d_n - c_{n1}x_1^{(k)} - \dots - c_{n,n-1}x_{n-1}^{(k)}.\end{aligned} \tag{3.6}$$

Это значит, что $x_1^{(k)}$ находится так же, как и в методе простой итерации, т. е. в правую часть первого уравнения вместо x_2, \dots, x_n подставляются значения $x_2^{(k-1)}, \dots, x_n^{(k-1)}$, полученные на предыдущей, $(k-1)$ -ой итерации. При вычислении $x_2^{(k)}$ в правую часть вместо x_1 подставляется только что найденное значение $x_1^{(k)}$, а величины x_3, \dots, x_n берутся из предыдущей, $(k-1)$ -ой итерации.

Обобщая, можно сказать, что в методе Зейделя значения $x_i^{(k)}$ используются для определения всех последующих величин $x_{i+1}^{(k)}, \dots, x_n^{(k)}$ на той же k -ой итерации.

При вычислениях по формулам (3.5) и (3.6) возникают следующие вопросы:

1) Как ведут себя величины $x_i^{(k)}$ по мере увеличения номера итерации k ?

2) На какой итерации следует остановить вычислительный процесс?

3) Что дает переход от формул (3.5) к формулам (3.6), т. е. от метода простой итерации к методу Зейделя?

4) Что произойдет, если попытаться решить итерационными методами систему, не имеющую диагонального преобладания?

5) Каковы преимущества и недостатки итерационных методов по сравнению с прямыми методами, такими, как метод Гаусса, и какова область применения итерационных методов?

Ниже даются краткие ответы на эти вопросы.

1. На первых итерациях изменение величин $x_i^{(k)}$ носит, как правило, хаотичный характер. Однако постепенно их значения стабилизируются: сначала устанавливается правильный знак числа, затем появляется и начинает повторяться первая правильная цифра, затем вторая и т. д. Это свидетельствует о том, что итерационный процесс сходится. Причем сходится независимо от того, с какого начального приближения он стартовал.

2. Процесс вычислений по формулам (3.5) и (3.6) является в принципе бесконечным. И остановить его надо в тот момент, когда будет получено решение с наперед заданной точностью

(m верных значащих цифр). Исследования показывают, что необходимое для этого число итераций зависит не только от требуемой точности результата, но и от свойств самой системы (большей или меньшей степени выраженности ее диагонального преобладания, порядка), а также величины свободных членов и начальных значений неизвестных. Однако формулы и методики определения минимального числа итераций сложны, особенно в случае метода Зейделя. Поэтому на практике лучше пользоваться более простыми приемами.

Если система невысокого порядка решается с использованием калькулятора с последовательной записью на бумаге всех значений $x_i^{(k)}$, то имеется возможность в процессе решения наблюдать за их поведением. При этом легко обнаруживается тот момент, когда все неизвестные в двух соседних итерациях совпадают друг с другом до $m + 1$ верных значащих цифр. Тогда, округлив $x_i^{(k)}$ до m цифр, гарантировано получают результат требуемой точности.

При счете на ПК следить визуально за ходом вычислений практически невозможно. Поэтому на каждой итерации надо выполнять проверку

$$\left| \frac{x_i^{(k)} - x_i^{(k-1)}}{x_i^{(k)}} \right| \leq \varepsilon, \quad (3.7)$$

где ε – некоторое малое, наперед заданное число. Вычислительный процесс прекращается, если условие (3.7) выполняется для всех i от 1 до n . Это условие не срабатывает в одном случае, а именно, когда $x_i^{(k)}$ окажется равным нулю. Чтобы предупредить этот случай, представим (3.7) в виде

$$\left| x_i^{(k)} - x_i^{(k-1)} \right| \leq \varepsilon \left| x_i^{(k)} \right|,$$

а затем добавим к правой части малую величину ε

$$\left| x_i^{(k)} - x_i^{(k-1)} \right| \leq \varepsilon \left| x_i^{(k)} \right| + \varepsilon. \quad (3.8)$$

Это условие будет «работать» при любом $x_i^{(k)}$. В частности, при $x_i^{(k)} = 0$ оно вырождается в следующее:

$$\left| x_i^{(k-1)} \right| \leq \varepsilon.$$

Что касается величины ε , то анализ показывает: для получения результата с m верными значащими цифрами надо принять ε не более, чем $5 \cdot 10^{-(m+2)}$.

3. Интуиция подсказывает, что немедленный ввод в вычисления найденного на этой же итерации значения неизвестного, как это предусматривают формулы (3.6), должен привести к ускорению сходимости. И практика подтверждает, что метод Зейделя требует, как правило, меньшего числа итераций для получения результата той же точности. Однако строгого доказательства этого факта пока нет.

4. Если попытаться решать итерационными методами систему, не имеющую диагонального преобладания, то вычислительный процесс будет, как правило, расходящимся. Это значит, что с увеличением номера итерации значения неизвестных не стабилизируются, а неограниченно возрастают со знаком плюс, или минус, или с переменной знака. Наконец, наступает момент переполнения памяти машины и происходит остановка выполнения программы.

Однако такой исход вовсе не обязателен. Дело в том, что условие диагонального преобладания является только достаточным. Кроме того, области сходимости методов простой итерации и Зейделя пересекаются, но полностью не совпадают. Поэтому при рассмотрении конкретной системы могут встретиться четыре случая:

- 1) система не решается обоими итерационными методами;
- 2) система решается только методом простой итерации;
- 3) система решается только методом Зейделя;
- 4) система решается обоими методами.

Как мы уже знаем, условие диагонального преобладания гарантирует четвертый случай.

5. Итак, область применения итерационных методов – решение, главным образом, систем с диагональным преобладанием. Это недостаток итерационных методов, причем довольно существенный. Хотя в расчетной практике такие системы встречаются довольно часто. Однако эти методы имеют и преимущества.

Например, они в отличие от прямых методов дают результаты заданной точности. Поэтому итерационные методы можно использовать для уточнения решения, полученного прямыми методами (при этом значения неизвестных, найденные прямым методом, берутся в качестве начального приближения для итерационного уточнения).

В практических задачах возможно появление такой ситуации, когда система имеет настолько высокий порядок, что ее невозможно расположить в оперативной памяти ПК, но уравнения очень простые. В этом случае их формирование можно непосредственно включить в итерационный процесс решения, что при применении прямых методов сделать невозможно.

3.2. Пример решения системы линейных алгебраических уравнений итерационными методами

Снова рассмотрим систему линейных уравнений третьего порядка

$$\begin{aligned} 25,2x_1 - 4,7x_2 + 1,4x_3 &= 46,8, \\ 4,1x_1 + 39,4x_2 + 2,9x_3 &= -25,6, \\ -1,3x_1 + 3,1x_2 + 16,8x_3 &= 34,3, \end{aligned} \quad (3.9)$$

которая уже была решена в лабораторной работе № 2 методом Гаусса. В настоящей работе найдем решение этой системы итерационными методами – методом простой итерации и методом Зейделя.

Решение.

1. Проверяем условие вырожденности системы. Так как определитель матрицы не равен 0, то решение системы существует и единственно.

2. Проверим условие диагонального преобладания.

$$|25,2| > |-4,7| + |1,4|; \quad |39,4| > |4,1| + |2,9|; \quad |16,8| > |-1,3| + |3,1|.$$

Так как условие выполняется, то для поиска решения можно использовать оба метода: простой итерации и Зейделя.

3. Преобразование заданной системы к нормальному виду.

Решая первое уравнение относительно x_1 , второе – относительно x_2 , а третье – относительно x_3 , получаем

$$\begin{aligned}x_1 &= 1,85714 + 0,186508x_2 - 0,0555556x_3, \\x_2 &= -0,649746 - 0,104061x_1 - 0,0736041x_3, \\x_3 &= 2,04167 + 0,0773810x_1 - 0,184524x_2.\end{aligned}\quad (3.10)$$

4. Задание начального приближения.

Мы уже имеем решение этой системы методом Гаусса. И его можно бы использовать в качестве начального приближения для итерационного уточнения решения. Однако, чтобы показать возможности итерационных методов, примем в качестве начальных значений числа, которые отличаются от искомым величин не только абсолютным значением, но и знаком: $x_1^{(0)} = -3$, $x_2^{(0)} = 2$, $x_3^{(0)} = 1$.

5. Проведение итераций.

Вначале применим *метод простой итерации*.

Подставив начальные значения в правую часть нормальной системы (3.10), получаем первое приближение

$$\begin{aligned}x_1^{(1)} &= 1,85714 + 0,186508 \cdot 2 - 0,0555556 \cdot 1 = 2,17460, \\x_2^{(1)} &= -0,649746 - 0,104061 \cdot (-3) - 0,0736041 \cdot 1 = -0,411168, \\x_3^{(1)} &= 2,04167 + 0,0773810 \cdot (-3) - 0,184524 \cdot 2 = 1,44048.\end{aligned}$$

Полученные значения вновь подставляем в правую часть нормальной системы и находим второе приближение

$$\begin{aligned}x_1^{(2)} &= 1,85714 + 0,186508 \cdot (-0,411168) - 0,0555556 \cdot 1,44048 = 1,70043, \\x_2^{(2)} &= -0,649746 - 0,104061 \cdot 2,17460 - 0,0736041 \cdot 1,44048 = -0,982062, \\x_3^{(2)} &= 2,04167 + 0,0773810 \cdot 2,17460 - 0,184524 \cdot (-0,411168) = 2,28581.\end{aligned}$$

Эти и последующие приближения неизвестных сведем в табл. 3.1, где величина k обозначает номер итерации или номер приближения.

Таблица 3.1

Решение системы (3.9) методом простой итерации

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	-3	2	1
1	2,17460	-0,411168	1,44048
2	1,70043	-0,982062	2,28581
3	1,54699	-0,994939	2,35446
4	1,54078	-0,984025	2,34396
5	1,54334	-0,982680	2,34247
6	1,54373	-0,982763	2,34242
7	1,54372	-0,982800	2,34246
8	1,54371	-0,982802	2,34247

Анализ этой таблицы показывает, как формируется решение в процесс сходимости: правильные знаки неизвестных устанавливаются уже на первой итерации, первая правильная цифра – на второй итерации, вторая – на четвертой итерации и т. д. Процесс вычислений остановлен в тот момент, когда у всех неизвестных повторились первые пять значащих цифр. Округлив эти значения до четырех цифр, получаем $x_1 = 1,5437$, $x_2 = -0,9828$, $x_3 = 2,3425$.

Найденные значения, вообще говоря, подлежат проверке подстановкой их во все уравнения заданной системы, как это делалось в лабораторной работе № 2. Однако факт совпадения полученных значений с решением, найденным методом Гаусса, тоже свидетельствуют о правильности результатов.

Решим теперь исходную систему по методу Зейделя, вычисляя приближения по формулам (3.6).

Имеем первое приближение:

$$x_1^{(1)} = d_1 - c_{12}x_2^{(0)} - c_{13}x_3^{(0)} = 1,85714 + 0,186508 \cdot 2 - 0,0555556 \cdot 1 = 2,17460,$$

$$\begin{aligned} x_2^{(1)} &= d_2 - c_{21}x_1^{(1)} - c_{23}x_3^{(0)} = \\ &= -0,649746 - 0,104061 \cdot 2,17460 - 0,0736041 \cdot 1 = -0,949641, \end{aligned}$$

$$\begin{aligned} x_3^{(1)} &= d_3 - c_{31}x_1^{(1)} - c_{32}x_2^{(1)} = \\ &= 2,04167 + 0,0773810 \cdot 2,17460 - 0,184524 \cdot (-0,949641) = 2,38517. \end{aligned}$$

Второе приближение

$$x_1^{(2)} = d_1 - c_{12}x_2^{(1)} - c_{13}x_3^{(1)} = 1,85714 + 0,186508 \cdot (-0,949641) - 0,0555556 \cdot 2,38517 = 1,54572,$$

$$x_2^{(2)} = d_2 - c_{21}x_1^{(2)} - c_{23}x_3^{(1)} = -0,649746 - 0,104061 \cdot 1,54752 - 0,0736041 \cdot 2,38517 = -0,986341,$$

$$x_3^{(2)} = d_3 - c_{31}x_1^{(2)} - c_{32}x_2^{(2)} = 2,04167 + 0,0773810 \cdot 1,54752 - 0,184524 \cdot (-0,986341) = 2,34342.$$

За поведением решения можно проследить по табл. 3.2.

Таблица 3.2

Решение системы (3.9) методом Зейделя

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	-3	2	1
1	2,17460	-0,949641	2,38517
2	1,54752	-0,986341	2,34342
3	1,54299	-0,982796	2,34247
4	1,54371	-0,982797	2,34247
5	1,54371	-0,982801	2,34247

Как и следовало ожидать, итерационный процесс сошелся быстрее, чем в методе простой итерации.

3.3. Выполнение индивидуальных заданий

Составить систему линейных алгебраических уравнений, выбрав коэффициенты и свободные члены из табл. 3.3 согласно шифру. Решить эту систему методом простой итерации и методом Зейделя с точностью $E = 10^{-4}$. Сделать проверку полученных результатов. Решить задание, используя систему *Mathematica*, и сравнить полученные результаты.

Таблица 3.3.

Исходные данные к заданию

Первая цифра шифра	Коэффициенты системы уравнений			Вторая цифра шифра	Свободные члены b_i	Третья цифра шифра	Величина c
	a_{i1}	a_{i2}	a_{i3}				
1	2	3	4	5	6	7	8
0	14,34 – c –4,157 –3,306	1,588 12,750 4,530	–2,164 2,949 9,760	0	0,670 –4,174 0,428	0	0,513
1	11,230 4,679 –8,417	1,478 15,208 + c 0,784	3,245 1,039 10,732	1	1,767 3,134 1,987	1	0,317
2	9,214 5,298 4,297	3,047 7,319 1,040	2,046 1,395 7,943	2	4,278 1,409 0,903 + c	2	1,783
3	23,729 8,860 3,495	10,395 – c 14,395 2,845	1,010 4,045 13,230	3	2,170 4,194 3,294	3	–3,134
4	5,204 1,054 + c 0,371	3,956 20,423 5,320	0,030 2,569 17,456	4	4,276 –3,167 1,328	4	0,451
5	–13,057 1,390 5,281	2,239 19,432 4,130 + c	1,945 4,853 19,634	5	13,045 2,056 –1,940	5	0,574
6	11,945 2,478 3,287	1,035 15,374 7,813	3,275 4,230 + c 13,743	6	2,874 4,934 5,397	6	0,323
7	14,023 2,056 1,392	0,456 12,385 4,439	1,493 + c 5,179 13,845	7	6,905 2,504 4,05	7	0,124
8	9,394 9,010 2,156	1,034 15,345 7,032 – c	0,638 4,945 19,045	8	3,905 5,234 1,043 + c	8	–0,394
9	13,056 –3,456 –5,170	0,045 – c 7,845 –6,134	–0,198 –0,345 15,390	9	3,560 – c 2,157 10,150	9	–1,489

ЛАБОРАТОРНАЯ РАБОТА № 4. НАХОЖДЕНИЕ НАИБОЛЬШЕГО ПО МОДУЛЮ СОБСТВЕННОГО ЗНАЧЕНИЯ И СООТВЕТСТВУЮЩЕГО ЕМУ СОБСТВЕННОГО ВЕКТОРА КВАДРАТНОЙ МАТРИЦЫ

Цель работы: изучить степенной итерационный метод и его модификации для определения наибольшего по модулю собственного значения λ_1 и соответствующего ему собственного вектора \vec{x}_1 квадратной матрицы.

Состав работы:

1. Изучение алгоритма степенного итерационного метода.
2. Пример определения λ_1 и \vec{x}_1 степенным методом;
3. Недостатки степенного итерационного метода;
4. Модификации степенного итерационного метода;
5. Пример определения λ_1 и \vec{x}_1 модифицированным степенным методом;
6. Поиск собственных значений и собственных векторов матрицы в системе *Mathematica*;
7. Выполнение индивидуальных заданий.

4.1. Нахождение λ_1 и \vec{x}_1 степенным методом

Пусть дана квадратная матрица \mathbf{A} произвольного порядка n . Требуется решить для нее с заданной точностью частичную проблему собственных значений по определению λ_1 и \vec{x}_1 , т. е. такого числа λ_1 и такого вектора \vec{x}_1 , которые удовлетворяют соотношению

$$\mathbf{A} \vec{x}_1 = \lambda_1 \vec{x}_1 \quad (4.1)$$

(как известно, аналогичным соотношениям удовлетворяют второе и последующие собственные значения: $\mathbf{A} \vec{x}_2 = \lambda_2 \vec{x}_2, \dots$).

Для решения этой задачи степенным итерационным методом нужно задаться начальным вектором $\vec{y}^{(0)}$ порядка n , имеющим

произвольные, но ненулевые компоненты. Затем строится итерационный процесс

$$\bar{y}^{(k)} = \mathbf{A} \bar{y}^{(k-1)} \quad (k = 1, 2, \dots - \text{номера итераций}),$$

в результате которого образуется последовательность векторов

$$\bar{y}^{(0)}, \bar{y}^{(1)} = \mathbf{A} \bar{y}^{(0)}, \bar{y}^{(2)} = \mathbf{A} \bar{y}^{(1)}, \dots$$

В ходе вычисления этих векторов можно находить все более уточняющееся значение λ_1 по формуле

$$\lambda_1^{(k)} = \frac{y_s^{(k)}}{y_s^{(k-1)}} \quad (k = 1, 2, \dots), \quad (4.2)$$

где $y_s^{(k)}$ – s -я (любая, $s = 1, \dots, n$) компонента вектора $\bar{y}^{(k)}$ на k -й итерации;

$y_s^{(k-1)}$ – соответствующая компонента вектора $\bar{y}^{(k-1)}$ на предыдущей итерации.

k -е приближение для первого собственного вектора можно брать как

$$\bar{x}_i^{(k)} = \bar{y}^{(k)} \quad (k = 1, 2, \dots). \quad (4.3)$$

Для доказательства формул (4.2) и (4.3) разложим начальный вектор $\bar{y}^{(0)}$ по собственным векторам матрицы \mathbf{A}

$$\bar{y}^{(0)} = a_1 \bar{x}_1 + a_2 \bar{x}_2 + \dots + a_n \bar{x}_n, \quad (4.4)$$

где a_i ($i = 1, \dots, n$) – некоторые числа, причем $a_1 \neq 0$.

Тогда $\bar{y}^{(1)} = \mathbf{A} \bar{y}^{(0)} = a_1 \mathbf{A} \bar{x}_1 + a_2 \mathbf{A} \bar{x}_2 + \dots + a_n \mathbf{A} \bar{x}_n$.

Исходя из определения собственного значения и собственного вектора

$$\mathbf{A} \bar{x}_1 = \lambda_1 \bar{x}_1, \mathbf{A} \bar{x}_2 = \lambda_2 \bar{x}_2, \dots, \mathbf{A} \bar{x}_n = \lambda_n \bar{x}_n. \quad (4.5)$$

Поэтому $\bar{y}^{(1)} = a_1 \lambda_1 \bar{x}_1 + a_2 \lambda_2 \bar{x}_2 + \dots + a_n \lambda_n \bar{x}_n$.

Переходя ко второму вектору, получаем

$$\bar{y}^{(2)} = \mathbf{A} \bar{y}^{(1)} = a_1 \lambda_1 \mathbf{A} \bar{x}_1 + a_2 \lambda_2 \mathbf{A} \bar{x}_2 + \dots + a_n \lambda_n \mathbf{A} \bar{x}_n$$

или

$$\bar{y}^{(2)} = a_1 \lambda_1^2 \bar{x}_1 + a_2 \lambda_2^2 \bar{x}_2 + \dots + a_n \lambda_n^2 \bar{x}_n. \quad (4.6)$$

Теперь очевидно, что вектор $\bar{y}^{(k)}$ ($k = 1, 2, \dots$) содержит в своем разложении собственные значения, возведенные в k -ю степень (отсюда и название рассматриваемого метода – степенной). Но поскольку λ_1 – наибольшее из всех n собственных значений матрицы, то доля первого слагаемого в разложениях (4.5), (4.6) и т. д. с увеличением номера итерации будет все более возрастать. Это дает основание в записанных выше и последующих разложениях отбросить второе, третье, ... , n -е слагаемое, записав приближенные равенства

$$\begin{aligned} \bar{y}^{(0)} &= a_1 \bar{x}_1, \\ \bar{y}^{(1)} &= a_1 \lambda_1 \bar{x}_1, \\ \bar{y}^{(2)} &= a_1 \lambda_1^2 \bar{x}_1, \\ &\dots \end{aligned} \quad (4.7)$$

Такие же зависимости справедливы и для отдельных компонентов входящих сюда векторов

$$\begin{aligned} y_s^{(0)} &= a_1 x_{s1}, \\ y_s^{(1)} &= a_1 \lambda_1 x_{s1}, \\ y_s^{(2)} &= a_1 \lambda_1^2 x_{s1}, \\ &\dots \end{aligned}$$

где s может принимать любое значение среди чисел $1, \dots, n$.

Разделив второе равенство на первое, получим λ_1 в первом приближении

$$\lambda_1^{(1)} = \frac{y_s^{(1)}}{y_s^{(0)}}.$$

Чтобы получить второе приближение для λ_1 , нужно разделить третье равенство на второе:

$$\lambda_1^{(2)} = \frac{y_s^{(2)}}{y_s^{(1)}}.$$

На любой k -й итерации имеем формулу (4.2).

Справедливость формулы (4.3) вытекает из приближенных равенств (4.7). При этом следует вспомнить, что собственный вектор остается таким же, будучи умноженным на любое число, отличное от нуля. Поскольку $a_1 \neq 0$, то на первой итерации за \bar{x}_1 можно взять вектор $a_1 \lambda_1 \bar{x}_1$ или \bar{y}_1 , на второй – $a_1 \lambda_1^2 \bar{x}_1$ или $\bar{y}^{(2)}$ и т. д.

Конечно, проводя вычисления по формулам (4.2) и (4.3), мы получим для λ_1 и \bar{x}_1 на первых итерациях самые немыслимые значения. Однако постепенно искомые величины будут стабилизироваться, пока не сойдутся с заданной точностью к своим истинным значениям, что можно проверить подстановкой в соотношение (4.1). Момент окончания счета определяется так же, как и в итерационных методах решения систем линейных алгебраических уравнений.

4.2. Пример определения λ_1 и \bar{x}_1 степенным методом

Пусть дана матрица третьего порядка

$$\mathbf{A} = \begin{bmatrix} 25,2 & -4,7 & 1,4 \\ 4,1 & 39,4 & 2,9 \\ -1,3 & 3,1 & 16,8 \end{bmatrix}. \quad (4.8)$$

Требуется определить λ_1 и \bar{x}_1 с точностью $\varepsilon = 10^{-3}$, используя степенной итерационный метод.

Решение.

Зададимся начальным вектором третьего порядка $\bar{y}^0 = [5 \quad -2 \quad 1]^T$ и вычислим

$$\bar{y}^{(1)} = \mathbf{A} \bar{y}^{(0)} = \begin{bmatrix} 25,2 & -4,7 & 1,4 \\ 4,1 & 39,4 & 2,9 \\ -1,3 & 3,1 & 16,8 \end{bmatrix} \cdot \begin{bmatrix} 5 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 136,8 \\ -55,4 \\ 4,1 \end{bmatrix}.$$

Примем $s = 2$. Тогда первое приближение для λ_1 определится как $\lambda_1^{(1)} = y_2^{(1)} / y_2^{(0)} = -55,4 / (-2) = 27,7$.

Соответствующий собственный вектор

$$\bar{x}_1^{(1)} = \bar{y}^{(1)} = [136,8 \quad -55,4 \quad 4,1]^T.$$

Переходим ко второй итерации:

$$\bar{y}^{(2)} = \mathbf{A} \bar{y}^{(1)} = \begin{bmatrix} 25,2 & -4,7 & 1,4 \\ 4,1 & 39,4 & 2,9 \\ -1,3 & 3,1 & 16,8 \end{bmatrix} \cdot \begin{bmatrix} 136,8 \\ -55,4 \\ 4,1 \end{bmatrix} = \begin{bmatrix} 3713,48 \\ -1609,99 \\ -280,7 \end{bmatrix},$$

$$\lambda_1^{(2)} = y_2^{(2)} / y_2^{(1)} = -1609,99 / (-55,4) = 29,0612,$$

$$\bar{x}_1^{(2)} = \bar{y}^{(2)} = [3713,48 \quad -1609,99 \quad -280,7]^T.$$

Продолжая итерационный процесс, сведем эти и последующие результаты в табл. 4.1.

Таблица 4.1

Нахождение λ_1 и \bar{x}_1 для матрицы (4.8) степенным итерационным методом

k	$y_1^{(k)}$	$y_2^{(k)}$	$y_3^{(k)}$	$\lambda_1^{(k)} =$ $= y_2^{(k)} / y_2^{(k-1)}$	$\varepsilon^{(k)} =$ $= \lambda_1^{(k)} - \lambda_1^{(k-1)} $
0	5	-2	1	-	-
1	$1,368 \cdot 10^2$	$-5,54 \cdot 10^1$	$4,1 \cdot 10^0$	27,7	-
2	$3,71348 \cdot 10^3$	$-1,60999 \cdot 10^3$	$-2,807 \cdot 10^2$	29,0612	1,3612

1	2	3	4	5	6
3	$1,00754 \cdot 10^5$	$-4,90224 \cdot 10^4$	$-1,45343 \cdot 10^4$	30,4489	1,3877
...
26	$2,29435 \cdot 10^4$ 0	$-6,81085 \cdot 10^{40}$	$-1,11122 \cdot 10^{40}$	38,4912	0,0012
27	$8,82730 \cdot 10^4$ 1	$-2,62163 \cdot 10^{42}$	$-4,27648 \cdot 10^{41}$	38,4920	0,0008

Вычисления прекращены на 27-й итерации, так как $\varepsilon^{(27)}$ меньше $\varepsilon = 10^{-3}$:

$$\lambda_1^{(27)} = 38,4920,$$

$$\bar{x}_1^{(27)} = \left[8,82730 \cdot 10^{41} \quad -2,62163 \cdot 10^{42} \quad -4,27648 \cdot 10^{41} \right]^T.$$

Проверим эти значения подстановкой в равенство (4.1):

$$A \bar{x}_1^{(27)} = \begin{bmatrix} 25,2 & -4,7 & 1,4 \\ 4,1 & 39,4 & 2,9 \\ -1,3 & 3,1 & 16,8 \end{bmatrix} \cdot \begin{bmatrix} 8,82730 \cdot 10^{41} \\ -2,62163 \cdot 10^{42} \\ -4,27648 \cdot 10^{41} \end{bmatrix} = \begin{bmatrix} 3,39678 \cdot 10^{43} \\ -1,00913 \cdot 10^{44} \\ -1,64591 \cdot 10^{43} \end{bmatrix},$$

$$\lambda_1^{(27)} \bar{x}_1^{(27)} = 38,4920 \cdot \begin{bmatrix} 8,82730 \cdot 10^{41} \\ -2,62163 \cdot 10^{42} \\ -4,27648 \cdot 10^{41} \end{bmatrix} = \begin{bmatrix} 3,39780 \cdot 10^{43} \\ -1,00912 \cdot 10^{44} \\ -1,64610 \cdot 10^{43} \end{bmatrix}.$$

Как видно, соответствующие компоненты левой и правой частей равенства (4.1) практически совпадают друг с другом. Таким образом, полученный результат следующий:

$$\lambda_1 = 38,492; \quad \bar{x}_1 = \left[8,8273 \cdot 10^{41} \quad -2,62163 \cdot 10^{42} \quad -4,27648 \cdot 10^{41} \right]^T.$$

Однако собственным вектором, представленным таким образом, пользоваться неудобно. Чаще всего его тем или иным способом **нормируют**. Наиболее употребительны два способа нормирования:

1) делением \bar{x}_1 на его наибольшую по модулю компоненту (в данном случае таковой является вторая компонента $-2,62163 \cdot 10^{42}$). Тогда в новом (нормированном) векторе соответствующая компонента станет равной +1, а остальные по модулю не будут превосходить единицу. Нормированный таким способом собственный вектор обозначим одной звездочкой: $\bar{x}_1^* = [-0,3367 \quad 1 \quad 0,1631]^T$;

2) делением \bar{x}_1 на его длину (модуль). Полученный таким способом вектор называется нормированным к единичной длине. Мы будем обозначать его двумя звездочками.

В нашей задаче

$$|\bar{x}^{(27)}| = \sqrt{(8,82730 \cdot 10^{41})^2 + (-2,62163 \cdot 10^{42})^2 + (-4,27648 \cdot 10^{41})^2} = 2,79912 \cdot 10^{42}$$

и

$$\bar{x}_1^{**} = [0,3154 \quad -0,9366 \quad -0,1528]^T.$$

Длина этого вектора равна 1.

Векторы \bar{x}_1 , \bar{x}_1^* , \bar{x}_1^{**} отличаются друг от друга на некоторый множитель, т. е. имеют разную длину, но одинаковое направление, поэтому каждый из них является собственным вектором заданной матрицы (4.8).

4.3. Недостатки степенного итерационного метода

Степенной итерационный метод обладает рядом существенных недостатков:

1) если $|\lambda_1| > 1$ (как это имеет место в нашем примере), то с увеличением номера итерации k компоненты вектора $\bar{y}^{(k)}$ могут возрасти настолько, что превысят верхнюю границу области допустимых значений для вещественных чисел и произойдет аварийная остановка программы при ее выполнении на ЭВМ;

2) если $|\lambda_1| < 1$, то с увеличением номера итерации k компоненты вектора $\bar{y}^{(k)}$ могут уменьшиться настолько, что они окажутся мень-

ше нижней границы области допустимых значений для вещественных чисел и будут заменены машинным нулем, что означает потерю информации;

3) В ходе итерационного процесса компоненты вектора $\vec{y}^{(k)}$ могут менять знак (как это случилось в нашем примере с третьей компонентой при переходе от первого приближения ко второму, см. табл. 4.1). В процессе перемены знака компонента может пройти через нуль, и тогда при делении, как это предусматривает формула (4.2), произойдет аварийная остановка программы. Эта же причина вынуждает отказаться от задания нулевых компонент в начальном векторе $\vec{y}^{(0)}$;

4) Сходимость процесса можно отслеживать не только по λ_1 . Между тем, компоненты собственного вектора могут иметь более медленную сходимость, что может быть установлено только при проверке результатов.

Проще всего устраняется третий недостаток. Для этого $\lambda_1^{(k)}$ надо вычислять как среднее из тех m частных $y_s^{(k)} / y_s^{(k-1)}$, у которых $y_s^{(k-1)} \neq 0$, т. е.

$$\lambda_1^{(k)} = \frac{1}{m} \sum_{s=1}^n \frac{y_s^{(k)}}{y_s^{(k-1)}}.$$

Здесь $m \leq n$, т. е. слагаемые, которым отвечает $y_s^{(k-1)} = 0$, пропускаются. Теперь к заданию компонент вектора $\vec{y}^{(0)}$ может использоваться менее жесткий подход, однако все они не могут быть одновременно равными нулю.

4.4. Модификации степенного итерационного метода

Существенные недостатки, присущие степенному итерационному методу, явились предпосылкой исследования новых путей его совершенствования. Так появились модифицированные степенные методы.

Самая простая модификация, ликвидирующая, однако, сразу все четыре недостатка степенного итерационного метода (п. 4.3), состоит в том, что последовательность векторов строится по формулам

$$\begin{aligned}\bar{y}^{(0)}, \quad \bar{x}^{(0)} &= \frac{1}{y_s^{(0)}} \bar{y}^{(0)}, \\ \bar{y}^{(1)} &= \mathbf{A} \bar{x}^{(0)}, \quad \bar{x}^{(1)} = \frac{1}{y_s^{(1)}} \bar{y}^{(1)}, \\ \bar{y}^{(2)} &= \mathbf{A} \bar{x}^{(1)}, \quad \bar{x}^{(2)} = \frac{1}{y_s^{(2)}} \bar{y}^{(2)}, \\ &\dots\dots\dots\end{aligned}$$

Здесь $y_s^{(k)}$ – наибольшая по модулю компонента вектора $\bar{y}^{(k)}$. Это значит, что на каждой итерации выполняется нормирование вектора первым способом, т. е. делением его на наибольшую по модулю компоненту. И этот нормированный вектор используется для нахождения следующего вектора $\bar{y}^{(k+1)}$.

При таком построении итерационного процесса формула (4.2) примет вид

$$\lambda_1^{(k)} = y_s^{(k)} / x_s^{(k-1)}, \quad (k = 1, 2, \dots), \quad (4.11)$$

где $y_s^{(k)}$ – наибольшая по модулю компонента вектора $\bar{y}^{(k)}$, а $x_s^{(k-1)}$ – соответствующая компонента вектора $\bar{x}^{(k-1)}$.

На одной-двух первых итерациях номер s компоненты, имеющей наибольший модуль, может меняться. Дальше s устанавливается и сохраняет свое значение, после чего на всех итерациях $x_s^{(k-1)}$ будет равен единице и формула (4.11) упрощается

$$\lambda_1^{(k)} = y_s^{(k)}, \quad (k = 1, 2, \dots). \quad (4.12)$$

Мы будем пользоваться формулой (4.12) и на первых итерациях, т.к. полученные на них значения λ_1 будут далекими от истинной величины независимо от того, по какой формуле их вычислять.

В качестве k -го приближения первого собственного вектора можно взять $\vec{x}^{(k)}$. Это будет нормированный первым способом вектор

$$\vec{x}_1^{(k)} = \vec{x}^{(k)}, \quad (k = 1, 2, \dots). \quad (4.13)$$

Можно построить и другой модифицированный степенной метод, нормируя векторы $\vec{y}^{(k)}$ вторым способом (делением на длину вектора $|\vec{y}^{(k)}|$). В этом случае строится последовательность векторов:

$$\begin{aligned} \vec{y}^{(0)}, \vec{x}^{(0)} &= \frac{1}{|\vec{y}^{(0)}|} \vec{y}^{(0)}, \\ \vec{y}^{(1)} &= A \vec{x}^{(0)}, \vec{x}^{(1)} = \frac{1}{|\vec{y}^{(1)}|} \vec{y}^{(1)}, \\ \vec{y}^{(2)} &= A \vec{x}^{(1)}, \vec{x}^{(2)} = \frac{1}{|\vec{y}^{(2)}|} \vec{y}^{(2)}, \\ &\dots \end{aligned} \quad (4.14)$$

При этом k -е приближение для первого собственного значения определяется по формуле

$$\lambda_1^{(k)} = (\vec{y}^{(k)}, \vec{x}^{(k-1)}), \quad (k = 1, 2, \dots),$$

что означает скалярное произведение двух векторов, равное, как известно, сумме произведений их соответствующих компонент, т. е.

$$\lambda_1^{(k)} = \sum_{i=1}^n y_i^{(k)} x_i^{(k-1)}, \quad (k = 1, 2, \dots). \quad (4.15)$$

k -е приближение для первого собственного вектора, нормированного к единице, находится как

$$\vec{x}_1^{**k} = \vec{x}^{(k)}, \quad (k = 1, 2, \dots). \quad (4.16)$$

4.5. Пример определения λ_1 и \bar{x}_1 модифицированным степенным методом

Рассмотрим снова матрицу (4.8) и вычислим для нее λ_1 и \bar{x}_1 модифицированным степенным методом с точностью $\varepsilon = 10^{-3}$.

Зададим тот же начальный вектор $\bar{y}^0 = [5 \quad -2 \quad 1]^T$. Его наибольшая по модулю компонента $y_1^{(0)} = 5$ и имеет номер $s = 1$. Тогда начальный нормированный вектор будет $\bar{x}^{(0)} = [1 \quad -0,4 \quad 0,2]^T$.

Выполним первую итерацию

$$\bar{y}^{(1)} = \mathbf{A} \bar{x}^{(0)} = \begin{bmatrix} 25,2 & -4,7 & 1,4 \\ 4,1 & 39,4 & 2,9 \\ -1,3 & 3,1 & 16,8 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -0,4 \\ 0,2 \end{bmatrix} = \begin{bmatrix} 27,36 \\ -11,08 \\ 0,82 \end{bmatrix},$$

$\lambda_1^{(1)} = y_1^{(1)} = 27,36$ (наибольшая компонента, $s = 1$).

Нормированный вектор $\bar{x}^{(1)} = [1 \quad -0,405 \quad 0,03]^T = \bar{x}_1^{*(1)}$.

Перейдем ко второй итерации

$$\bar{y}^{(2)} = \mathbf{A} \bar{x}^{(1)} = \begin{bmatrix} 25,2 & -4,7 & 1,4 \\ 4,1 & 39,4 & 2,9 \\ -1,3 & 3,1 & 16,8 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -0,405 \\ 0,003 \end{bmatrix} = \begin{bmatrix} 27,1453 \\ -11,7689 \\ -2,0519 \end{bmatrix},$$

$\lambda_1^{(2)} = y_1^{(2)} = 27,1453$.

Нормированный вектор $\bar{x}^{(2)} = [1 \quad -0,4336 \quad -0,0756]^T = \bar{x}_1^{*(2)}$.

Эти и все последующие значения на итерациях сведены в табл. 4.2.

Таблица 4.2

Нахождение λ_1 и \bar{x}_1 для матрицы (4.8) модифицированным степенным методом

k	$y_1^{(k)}$	$y_2^{(k)}$	$y_3^{(k)}$	$\lambda_1^{(k)} = y_{\max}^{(k)}$	$\varepsilon^{(k)} = \lambda^{(k)} - \lambda^{(k-1)} $	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	5	-2	1			1	-0,4	0,2
1	27,36	-11,08	0,82	27,36		1	-0,405	0,03
2	27,1453	-11,7689	-2,0519	27,1453	0,2147	1	-0,4336	-0,0756
3	27,1319	-13,2012	-3,9139	27,1319	0,0134	1	-0,4866	-0,1443
...
6	28,0866	-23,4296	-7,2375	28,0866	0,4871	1	-0,8342	-0,2577
7	28,7599	-29,5144	-8,2151	28,7599	0,6733	-0,9744	1	0,2783
8	-28,8662	36,2120	9,0429	36,2120	7,4521	-0,7971	1	0,2497
...
26	-12,9664	38,4912	6,2800	38,4912	0,0012	-0,3369	1	0,1632
27	-12,9606	38,4920	6,2789	38,4920	0,0008	-0,3367	1	0,1631

Следует отметить, что наибольшая компонента на 7-й итерации поменяла номер (было $s = 1$, стало $s = 2$).

Остановив вычисления при $\varepsilon^{(k)} \leq \varepsilon$ (это 27-я итерация), получаем ответ, аналогичный как и при решении предыдущим способом

$$\lambda_1 = y_2^{(27)} = 38,492; \bar{x}_1^* = \bar{x}^{(27)} = [-0,3367 \quad 1 \quad 0,1631]^T.$$

4.6. Поиск собственных значений и собственных векторов матрицы в системе Mathematica

Собственные значения и собственные вектора матриц можно найти, используя функции **Eigenvalues** и **Eigenvectors**. Для того, чтобы определить характеристический полином матрицы, используется функция **CharacteristicPolynomial**.

Реализация примера из п. 4.2 в системе *Mathematica* показана на рисунке.

Функции **Eigenvalues** и **Eigenvectors** возвращают все собственные значения и соответствующие им собственные вектора. Значение рассчитанного в системе Mathematica наибольшего по модулю значения λ равно 38,4937, что практически совпадает с найденным с помощью итерационных методов. Значения компонент собственного вектора также практически совпадают (см. п. 4.2 при использовании нормировки на длину вектора).

```

A =  $\begin{pmatrix} 25.2 & -4.7 & 1.4 \\ 4.1 & 39.4 & 2.9 \\ -1.3 & 3.1 & 16.8 \end{pmatrix}$ ;
Eigenvalues[A]
Eigenvectors[A]
{38.4937, 26.1007, 16.8056}
{{0.315084, -0.936693, -0.152734},
{-0.945674, 0.244915, 0.213814},
{-0.210923, -0.0866941, 0.973651}}
CharacteristicPolynomial[A, x]
16884.8 - 2090.26 x + 81.4 x2 - x3

```

4.7. Выполнение индивидуальных заданий

Составить матрицу, выбрав коэффициенты из табл. 4.3 согласно шифру. Используя степенной итерационный метод и его модификацию, определить наибольшее по модулю собственное значение λ (точность $\varepsilon = 10^{-3}$) и соответствующий ему собственный вектор. Провести нормировку собственного вектора двумя способами. Сделать проверку полученных результатов. Решить задание, используя систему *Mathematica*.

Таблица 4.3

Исходные данные к заданию

Первая цифра шифра	Коэффициенты системы уравнений			Вторая цифра шифра	Свободные члены b_i	Третья цифра шифра	Величина c
	a_{i1}	a_{i2}	a_{i3}				
1	2	3	4	5	6	7	8
0	1,034 15,345 + c 2,056	2,874 - d 0,456 5,204	2,504 11,945 13,056	0	0,124	0	1,783
1	4,157 12,750 - d 8,860	0,371 4,218 1,390	0,030 3,275 + c 1,390	1	0,574	1	0,030

Окончание табл. 4.3

1	2	3	4	5	6	7	8
2	1,588 + <i>c</i> 14,395 1,390	4,945 4,276 3,495	13,045 1,035 5,179 + <i>d</i>	2	1,040	2	-1,940
3	14,023 2,478 2,239 + <i>c</i>	5,204 5,281 + <i>d</i> 5,32	2,874 9,010 17,456	3	0,323	3	-0,394
4	13,057 5,204 + <i>d</i> -3,456	15,345 5,397 3,287	7,813 1,430 + <i>c</i> 4,130	4	-0,198	4	1,588
5	0,371 + <i>c</i> 12,385 2,949	4,853 13,729 - <i>d</i> 4,045	2,845 -0,345 1,034	5	0,638	5	0,317
6	4,230 9,394 2,156 - <i>d</i>	3,956 + <i>c</i> 15,374 -6,134	1,493 3,245 -4,157	6	0,451	6	1,054
7	8,860 11,945 19,045	19,432 7,032 + <i>c</i> 2,056	0,456 7,319 + <i>d</i> 19,634	7	-1,489	7	1,395
8	2,056 2,569 + <i>c</i> 12,750	4,439 7,845 13,230	0,045 - <i>d</i> 13,845 3,275	8	1,010	8	-3,134
9	2,046 5,170 4,194 + <i>c</i>	13,056 4,934 20,423	1,392 1,945 - <i>d</i> 4,679	9	0,513	9	0,039

ЛАБОРАТОРНАЯ РАБОТА № 5 РЕШЕНИЕ ЗАДАЧИ КОШИ МЕТОДАМИ РУНГЕ-КУТТА

Цель работы: изучить методы Рунге-Кутта решения одного дифференциального уравнения первого порядка, систем дифференциальных уравнений первого порядка, а также уравнений второго и старших порядков.

Состав работы:

1. Общие сведения о дифференциальных уравнениях;
2. Решение задачи Коши для одного дифференциального уравнения первого порядка;
3. Пример решения задачи Коши для одного дифференциального уравнения первого порядка;
4. Применение методов Рунге-Кутта к решению систем дифференциальных уравнений первого порядка;
5. Пример решения задачи Коши для системы двух линейных дифференциальных уравнений первого порядка;
6. Применение методов Рунге-Кутта к решению дифференциальных уравнений второго и старшего порядков;
7. Решение задачи Коши в системе *Mathematica*.
8. Выполнение индивидуальных заданий.

5.1. Общие сведения о дифференциальных уравнениях

Обыкновенными дифференциальными уравнениями называются уравнения, содержащие одну независимую переменную, искомую функцию $y = y(x)$ и одну или несколько ее производных. Например, $F(x, y, y', \dots, y^{(n)}) = 0$. Здесь x – независимая переменная, $y^{(i)}$, $i = 1, 2, \dots, n$ – i -я производная неизвестной функции $y(x)$; F – некоторая функция от переменных x, y, y', y'', \dots .

Порядок дифференциального уравнения определяется наивысшим порядком n входящей в уравнение производной. Если в уравнении старшую производную выразить через производные более низкого порядка и независимую переменную, то получаются уравнения, разрешенные относительно старшей производной. Например, $y''' = f(x, y, y', y'')$.

Дифференциальные уравнения бывают линейные и нелинейные. Линейным называется уравнение, в котором неизвестная функция

и ее производные входят в первой степени. Например, $y'' = 2xy + x^2$. Остальные уравнения называются нелинейными.

Решением дифференциального уравнения называется произвольная функция $\varphi(x)$, которая при подстановке в уравнение превращает его в тождество.

Общее решение обыкновенного дифференциального уравнения n -го порядка, как известно, содержит n произвольных постоянных c_1, c_2, \dots, c_n . Частное решение получается из общего путем задания определенных значений произвольным постоянным $c_i, i = 1, 2, \dots, n$. Для выделения частного решения из общего нужно задать столько дополнительных условий, сколько произвольных постоянных содержится в общем решении, т. е. каков порядок дифференциального уравнения.

Для системы n обыкновенных дифференциальных уравнений первого порядка в качестве дополнительных условий задаются значения искомых функций и их производные при некоторых x .

В зависимости от способа задания дополнительных условий для получения частного решения дифференциального уравнения различают два типа задач: задача Коши и краевые задачи.

Если дополнительные условия задаются в одной точке x , то задача называется *задачей Коши*, а дополнительные условия называются *начальными условиями*. Точка $x = x_0$, в которой они задаются, – *начальной точкой*. Для уравнения первого порядка задается одно начальное условие $y(x_0) = y_0$, т. е. начальное значение искомой функции. Для уравнения второго порядка задаются два начальных условия $y(x_0) = y_0$ и $y'(x_0) = y_0'$, т. е. начальное значение искомой функции и ее первой производной. В общем случае для уравнения n -го порядка задаются: $y(x_0) = y_0, y'(x_0) = y_0', \dots, y^{(n-1)}(x_0) = y_0^{(n-1)}$.

Обычно задача Коши связана с изучением процессов, протекающих во времени (колебания строительных конструкций, процесс распределения тепла и т. д.). Поэтому аргумент в таких задачах (время) обозначается через t , а производные записываются с использованием точки.

Если дополнительные условия задаются в нескольких точках, то есть при разных x , то такая задача называется *краевой*, а дополнительные условия – *краевыми* или *граничными условиями*. Обычно граничные условия задаются в точках $x = a$ и $x = b$, являющихся границами области решения дифференциального уравнения.

5.2. Решение задачи Коши для одного дифференциального уравнения первого порядка

Пусть имеется задача Коши

$$\dot{y} = f(t, y), y(t_0) = y_0.$$

Требуется решить ее численно, т. е. найти значения y_1, y_2, \dots при значениях аргумента t_1, t_2, \dots , отстоящих друг от друга на шаг Δt . Это означает, что

$$t_1 = t_0 + \Delta t; t_2 = t_1 + \Delta t, \dots$$

Все многообразие методов решения этой задачи делится на две группы – одношаговые и многошаговые. В одношаговых методах величина y_i определяется по информации, имеющейся в предыдущем узле $i - 1$. Соответственно, в многошаговых методах используется информация в k предыдущих узлах ($k \geq 2$). Точность многошаговых методов, как правило, выше, но реализовать их сложнее. Поэтому в расчетной практике используют главным образом одношаговые методы.

Самым простым и исторически первым является *метод ломаных Леонардо Эйлера*. Вместе с тем, это и самый неточный метод. В нем очередное значение искомой функции определяется по формуле

$$y_i = y_{i-1} + k_{i-1}\Delta t, (5.1)$$

где k_{i-1} – тангенс угла наклона касательной к графику искомой функции в точке (t_{i-1}, y_{i-1}) . Чтобы его определить, надо величины t_{i-1} и y_{i-1} подставить в правую часть решаемого уравнения соответственно вместо t и y , т. е. $k_{i-1} = f(t_{i-1}, y_{i-1})$.

Геометрическая интерпретация метода показана на рис. 5.1.

К интегральной кривой, проходящей через точку A с координатами t_{i-1} и y_{i-1} , проведена касательная AB . Тангенс угла наклона ее к горизонту α_{i-1} обозначен через k_{i-1} .

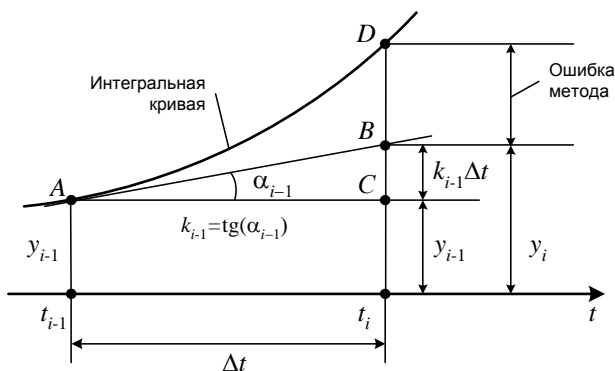


Рис. 5.1. Геометрическая интерпретация метода Эйлера

Очередное значение искомой функции y_i складывается из предыдущего значения y_{i-1} и приращения BC , которое можно определить следующим образом

$$|BC| = |AC| \operatorname{tg}(\alpha_{i-1}) = k_{i-1} \Delta t.$$

Из рисунка видно, что найденное значение y_i отличается от истинного значения на величину BD , являющуюся ошибкой метода, и наряду с ошибкой округления делающую результат приближенным. Ошибки способны накапливаться, снижая точность решения с каждым шагом.

Единственный способ повысить точность решения в методе Эйлера – уменьшить шаг. Именно это обстоятельство побудило немецких математиков К. Рунге и В. Кутта разработать целую гамму одношаговых методов, отличающихся порядком. Сюда входят: метод первого порядка (полностью совпадает с методом Эйлера), метод второго порядка и т. д. Чем выше порядок метода, тем точнее результат (при одном и том же шаге), но тем больше арифметических действий надо выполнить.

В *методе Рунге-Кутта второго порядка* вначале делается пробный шаг в узел i , как и методе первого порядка:

$$\begin{aligned} k_{i-1} &= f(t_{i-1}, y_{i-1}), \\ y_i^* &= y_{i-1} + k_{i-1} \Delta t. \end{aligned}$$

Затем находят тангенс угла α_i^* наклона касательной к интегральной кривой в точке (t_i, y_i^*)

$$k_i^* = f(t_i, y_i^*)$$

и определяют более точное значение искомой функции

$$y_i = y_{i-1} + 1/2 \cdot (k_{i-1} + k_i^*) \cdot \Delta t \quad (5.2)$$

(касательная из точки А проводится под углом, тангенс которого равен полусумме тангенсов k_{i-1} и k_i^*).

В настоящее время широкое применение в расчетной практике находит также метод Рунге-Кутты четвертого порядка.

5.3. Пример решения задачи Коши для одного дифференциального уравнения первого порядка

Решить методами первого и второго порядка задачу Коши

$$\dot{y} = 2y/t, \quad y(1) = 0,5 \quad (5.3)$$

на отрезке $[1, 2]$ в узлах с шагом $\Delta t = 0,2$.

Эта задача имеет точное аналитическое решение $y(t) = 0,5t^2$, которое позже используем для сравнения с приближенными результатами.

Решим эту задачу *методом Эйлера*.

Имеем:

1. Для первого узла ($i = 1$)

$$k_0 = 2 \frac{y_0}{t_0} = 2 \cdot \frac{0,5}{1} = 1; \quad y_1 = y_0 + k_0 \Delta t = 0,5 + 1 \cdot 0,2 = 0,7;$$

2. Для второго узла ($i = 2$)

$$k_1 = 2 \frac{y_1}{t_1} = 2 \cdot \frac{0,7}{1,2} = 1,16667;$$

$$y_2 = y_1 + k_1 \Delta t = 0,7 + 1,16667 \cdot 0,2 = 0,933333.$$

Аналогично проводятся вычисления для остальных узлов. Полученные результаты показаны в табл. 5.1.

Таблица 5.1

Точное и приближенное решение задачи Коши (5.3)

i	t_i	Аналитическое решение $y(t_i)$	Метод Эйлера y_i	Метод Рунге-Кутты второго порядка y_i
0		0,5	0,5	0,5
1	1,2	0,72	0,7	0,716667
2	1,4	0,98	0,933333	0,972620
3	1,6	1,28	1,2	1,26788
4	1,8	1,62	1,5	1,60246
5	2	2	1,83333	1,97636

Сравнивая их с результатами аналитического расчета, можно заключить, что приближенные значения содержат существенную погрешность, нарастающую по мере увеличения t_i . Для того, чтобы получить этим методом достаточно точное решение, нам потребуется значительно уменьшить шаг.

Решим теперь эту задачу *методом Рунге-Кутты второго порядка*.

1. Для первого узла ($i = 1$)

$$k_0 = 2 \frac{y_0}{t_0} = 2 \cdot \frac{0,5}{1} = 1; \quad y_1^* = y_0 + k_0 \cdot \Delta t = 0,5 + 1 \cdot 0,2 = 0,7,$$

$$k_1^* = 2 \frac{y_1^*}{t_1} = 2 \cdot \frac{0,7}{1,2} = 1,16667,$$

$$y_1 = y_0 + \frac{1}{2} (k_0 + k_1^*) \cdot \Delta t = 0,5 + \frac{1}{2} (1 + 1,16667) \cdot 0,2 = 0,716667;$$

2. Для второго узла ($i = 2$)

$$k_1 = 2 \frac{y_1}{t_1} = 2 \cdot \frac{0,716667}{1,2} = 1,19445,$$

$$y_2^* = y_1 + k_1 \cdot \Delta t = 0,716667 + 1,19445 \cdot 0,2 = 0,955557,$$

$$k_2^* = 2 \frac{y_2^*}{t_2} = 2 \cdot \frac{0,955557}{1,4} = 1,36508,$$

$$y_2 = y_1 + \frac{1}{2} (k_1 + k_2^*) \cdot \Delta t = 0,716667 + \frac{1}{2} (1,19445 + 1,36508) \cdot 0,2 = 0,972620.$$

Аналогично рассчитываются значения y_i в остальных узлах (табл. 5.1). Анализируя полученные результаты, можно заключить, что погрешность решения при использовании метода Рунге-Кутты второго порядка существенно уменьшилась.

5.4. Применение методов Рунге-Кутты к решению систем дифференциальных уравнений первого порядка

Пусть имеется задача Коши для системы n дифференциальных уравнений первого порядка

$$\begin{aligned} \dot{y} &= f(t, y, z, \dots), y(t_0) = y_0 \\ \dot{z} &= g(t, y, z, \dots), z(t_0) = z_0 \\ &\dots\dots\dots, \end{aligned} \tag{5.4}$$

где t – независимая переменная или аргумент; $y = y(t)$, $z = z(t)$, ... – искомые функции аргумента t .

Решая систему n дифференциальных уравнений с n неизвестными функциями, мы находимся в пространстве $(n + 1)$ измерений. Положение точки в таком пространстве определяется $(n + 1)$ числами – моментом времени t и значениями функций y и z . Через каждую точку пространства проходит n интегральных кривых $y = y(t)$, $z = z(t)$, К каждой из них можно провести свою касательную и определить тангенсы углов наклона через правые части соответствующих уравнений: $k = f(t, y, z, \dots)$, $l = g(t, y, z, \dots)$,

Остается только обобщить формулы (5.1) и (5.2).

Получаем:

1) для метода Эйлера (Рунге-Кутты первого порядка)

$$\begin{aligned} y_i &= y_{i-1} + k_{i-1}\Delta t, \\ z_i &= z_{i-1} + l_{i-1}\Delta t \quad (i = 1, 2, \dots), \\ &\dots\dots\dots, \end{aligned} \tag{5.5}$$

где $k_{i-1} = f(t_{i-1}, y_{i-1}, z_{i-1}, \dots)$ – тангенс угла наклона касательной к графику y в точке $(t_{i-1}, y_{i-1}, z_{i-1}, \dots)$; $l_{i-1} = g(t_{i-1}, y_{i-1}, z_{i-1}, \dots)$ – то же к графику z .

2) для метода Рунге-Кутты второго порядка

$$\begin{aligned} y_i &= y_{i-1} + 1/2(k_{i-1} + k_i^*)\Delta t \\ z_i &= z_{i-1} + 1/2(l_{i-1} + l_i^*)\Delta t, \quad (i = 1, 2, \dots), \\ &\dots\dots\dots, \end{aligned} \tag{5.6}$$

где k_{i-1} и l_{i-1} – то же, что и в формуле (5.5);

$k_i^* = f(t_i, y_i^*, z_i^*, \dots)$ – тангенс угла наклона касательной к графику y в точке $(t_i, y_i^*, z_i^*, \dots)$, причем $y_i^* = y_{i-1} + k_{i-1}\Delta t$, $z_i^* = z_{i-1} + l_{i-1}\Delta t$;
 $l_i^* = g(t_i, y_i^*, z_i^*, \dots)$ – то же, что и k_i^* , но к графику z .

Формулы численного интегрирования как отдельных дифференциальных уравнений, так и их систем не являются точными. На каждом шаге вычислительного процесса они дают некоторую погрешность, способную накапливаться и зависящую от порядка метода, величины шага, свойств самого решения. Поэтому встает вопрос не просто о численном решении задачи Коши, а о получении результата заданной точности.

Существует несколько подходов к решению этой проблемы. Самый простой, но и самый трудоемкий подход состоит в следующем: пусть требуется получить решение задачи Коши для дифференциального уравнения на отрезке $[t_0, t_k]$ с m верными значащими цифрами. Разобьем отрезок $[t_0, t_k]$ на две части, найдем шаг $\Delta t = (t_k - t_0)/2$ и зафиксируем на отрезке узлы t_1 и $t_2 = t_k$. Решив задачу Коши методом первого или второго порядка, получим значения $y_1^{(0)}$ и $y_2^{(0)}$. После этого уменьшим шаг вдвое и, зафиксировав узлы $t_1, t_2, t_3, t_4 = t_k$, снова решим ту же задачу и получим значения $y_1^{(1)}, y_2^{(1)}, y_3^{(1)}, y_4^{(1)}$. Проверяем условие

$$\left| y_4^{(1)} - y_2^{(0)} \right| \leq \varepsilon \left| y_4^{(1)} \right| + \varepsilon.$$

Это значит, что относительное расхождение между решениями на конце отрезка сравнивается с малой величиной, которую можно взять, например, равной $5 \cdot 10^{-(m+1)}$, где m – требуемое количество значащих цифр. Если это условие выполняется, то решение

$y_1^{(1)}, y_2^{(1)}, y_3^{(1)}, y_4^{(1)}$ можно считать искомым. В противном случае шаг Δt вновь уменьшается вдвое, из решения задачи находятся $y_1^{(2)}, \dots, y_8^{(2)}$, проверяется условие $|y_8^{(2)} - y_4^{(1)}| \leq \varepsilon |y_8^{(2)}| + \varepsilon$ и т. д.

5.5. Пример решения задачи Коши для системы двух линейных дифференциальных уравнений первого порядка

Пусть требуется решить задачу Коши

$$\begin{aligned} \dot{y} &= yt + z, y(0) = 1, \\ \dot{z} &= t + y, z(0) = -0,2, \end{aligned} \quad (5.7)$$

получив значения y_i, z_i с четырьмя верными значащими цифрами на отрезке $[0, 0,5]$ с шагом $\Delta t = 0,1$.

Для первых двух узлов имеем:

Метод Эйлера (Рунге-Кутта первого порядка)

1. Для первого узла ($i = 1$)

$$\begin{aligned} k_0 &= f(t_0, y_0, z_0) = t_0 y_0 + z_0 = 0 \cdot 1 - 0,2 = -0,2, \\ l_0 &= g(t_0, y_0, z_0) = t_0 + y_0 = 0 + 1 = 1, \\ y_1 &= y_0 + k_0 \Delta t = 1 - 0,2 \cdot 0,1 = 0,98, \\ z_1 &= z_0 + l_0 \Delta t = -0,2 + 1 \cdot 0,1 = -0,1; \end{aligned}$$

2. Для второго узла ($i = 2$)

$$\begin{aligned} k_1 &= f(t_1, y_1, z_1) = t_1 y_1 + z_1 = 0,1 \cdot 0,98 - 0,1 = -0,002, \\ l_1 &= g(t_1, y_1, z_1) = t_1 + y_1 = 0,1 + 0,98 = 1,08, \\ y_2 &= y_1 + k_1 \Delta t = 0,98 - 0,002 \cdot 0,1 = 0,9798, \\ z_2 &= z_1 + l_1 \Delta t = -0,1 + 1,08 \cdot 0,1 = 0,008. \end{aligned}$$

Метод Рунге-Кутта второго порядка

1. Для первого узла ($i = 1$)

$$\begin{aligned} k_0 &= f(t_0, y_0, z_0) = t_0 y_0 + z_0 = 0 \cdot 1 - 0,2 = -0,2, \\ l_0 &= g(t_0, y_0, z_0) = t_0 + y_0 = 0 + 1 = 1, \\ y_1^* &= y_0 + k_0 \Delta t = 1 - 0,2 \cdot 0,1 = 0,98, \\ z_1^* &= z_0 + l_0 \Delta t = -0,2 + 1 \cdot 0,1 = -0,1, \end{aligned}$$

$$\begin{aligned}
 k_1^* &= f(t_1, y_1^*, z_1^*) = t_1 y_1^* + z_1^* = 0,1 \cdot 0,98 - 0,1 = -0,002, \\
 l_1^* &= g(t_1, y_1^*, z_1^*) = t_1 + y_1^* = 0,1 + 0,98 = 1,08, \\
 y_1 &= y_0 + 1/2 \cdot (k_0 + k_1^*) \Delta t = 1 + 1/2 \cdot (-0,2 - 0,002) \cdot 0,1 = 0,9899, \\
 z_1 &= z_0 + 1/2 \cdot (l_0 + l_1^*) \Delta t = -0,2 + 1/2 \cdot (1 + 1,08) \cdot 0,1 = -0,096;
 \end{aligned}$$

2. Для второго узла ($i = 2$)

$$\begin{aligned}
 k_1 &= f(t_1, y_1, z_1) = t_1 y_1 + z_1 = 0,1 \cdot 0,9899 - 0,096 = 0,00299, \\
 l_1 &= g(t_1, y_1, z_1) = t_1 + y_1 = 0,1 + 0,9899 = 1,0899, \\
 y_2^* &= y_1 + k_1 \Delta t = 0,9899 + 0,00299 \cdot 0,1 = 0,990199, \\
 z_2^* &= z_1 + l_1 \Delta t = -0,096 + 1,0899 \cdot 0,1 = 0,01299, \\
 k_2^* &= f(t_2, y_2^*, z_2^*) = t_2 y_2^* + z_2^* = 0,2 \cdot 0,990199 + 0,01299 = 0,21103, \\
 l_2^* &= g(t_2, y_2^*, z_2^*) = t_2 + y_2^* = 0,2 + 0,990199 = 1,190199, \\
 y_2 &= y_1 + 1/2 \cdot (k_1 + k_2^*) \Delta t = 0,9899 + 1/2 \cdot (0,00299 + 0,211032) \cdot 0,1 = 1,000601, \\
 z_2 &= z_1 + 1/2 \cdot (l_1 + l_2^*) \Delta t = -0,096 + 1/2 \cdot (1,0899 + 1,190199) \cdot 0,1 = 0,018005.
 \end{aligned}$$

Значения y_i и z_i в остальных узлах рассчитываются аналогично. Полученные результаты показаны в табл. 5.2 и на рис. 5.2. Как видно, значения, полученные двумя методами, отличаются между собой и, конечно же, от истинных значений.

Таблица 5.2

Точное и приближенное решение задачи Коши (5.7)

i	t_i	Метод Эйлера		Метод Рунге-Кутты второго порядка	
		y_i	z_i	y_i	z_i
0	0	1	-0,2	1	-0,2
1	0,1	0,98	-0,1	0,9899	-0,096
2	0,2	0,9798	0,008	1,000601	0,018005
3	0,3	1,000196	0,12598	1,033747	0,144156
4	0,4	1,0428	0,256	1,091921	0,284802
5	0,5	1,110112	0,40028	1,178801	0,442602

Повысить точность полученного решения можно, уменьшив шаг. Однако это потребует огромной вычислительной работы, особенно для метода первого порядка. Для сокращения числа арифметических действий надо стремиться к использованию метода более

высокого порядка. Хотя на каждом шаге он требует выполнения большего числа действий, их общее количество для достижения результата той же точности оказывается меньшим. Еще большего снижения трудоемкости можно добиться при использовании метода четвертого порядка (необходимые формулы можно найти в литературе по численным методам).

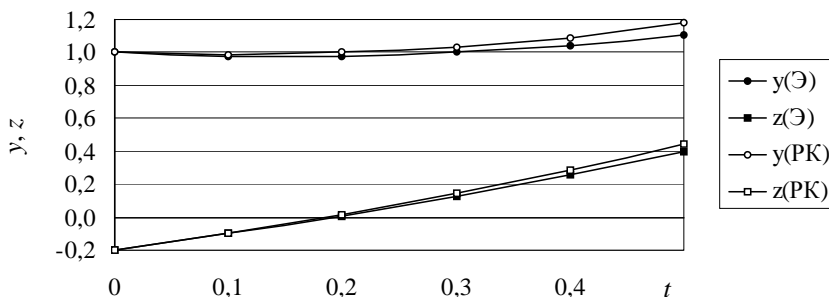


Рис. 5.2. Интегральные кривые искомых функций в задаче Коши (5.7)

5.6. Применение методов Рунге-Кутты к решению дифференциальных уравнений второго и старшего порядков

Как следует из вышеизложенного, методы Рунге-Кутты предназначены для решения отдельных дифференциальных уравнений первого порядка и систем таких уравнений. Если же возникает необходимость в решении уравнений второго или старших порядков, то исходную задачу Коши надо предварительно преобразовать, заменив уравнение n -го порядка с одной неизвестной функцией –

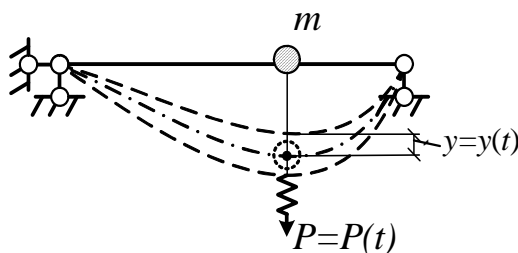


Рис. 5.3. Динамическая система с одной степенью свободы

системой n уравнений первого порядка с n неизвестными функциями. Соответственно этому изменяются и начальные условия.

В качестве *первого примера* рассмотрим задачу Коши, которой описываются вынужденные колебания системы с одной степе-

ню свободы. Пусть в произвольной точке однопролетной балки (рис. 5.3) установлен электромотор, деревообрабатывающий станок либо иное механическое устройство с массой m . От веса этого устройства балка изгибается, занимая положение статического равновесия, показанное на рис. 5.3 штрих-пунктирной линией. Когда устройство работает, оно создает динамическую нагрузку $P = P(t)$, под действием которой масса (а с ней и балка) совершает колебания вверх-вниз около положения статического равновесия (штриховые линии).

Масса балки обычно мала по сравнению с массой установленного на ней устройства. Пренебрегая ею, мы получаем систему с одной степенью свободы. Из динамики известно, что вынужденные колебания такой системы описываются следующей задачей Коши:

$$\begin{aligned} m\ddot{y} + b\dot{y} + ry &= P(t), \\ y(t_0) = y_0, \quad \dot{y}(t_0) &= \dot{y}_0, \end{aligned} \quad (5.8)$$

где m – масса устройства, кг;

b – коэффициент сопротивления, величина которого зависит, главным образом, от того, в какой среде происходят колебания (воздухе, воде, машинном масле и т. д.), кг/с;

r – коэффициент жесткости балки в точке расположения устройства; численно равен той силе, которую нужно приложить вместо массы, чтобы балка в той точке получила единичное перемещение, кг/см²;

$P(t)$ – динамическая нагрузка – сила, меняющаяся во времени свою величину по некоторому известному закону H ;

$y_0, \dot{y}(0)$ – начальное перемещение (м) и начальная скорость (м/с) массы, т. е. перемещение и скорость в момент времени $t = t_0$.

Таким образом, имеем задачу Коши для уравнения второго порядка. Ее решение следует начать с преобразования. Для этого вводится обозначение $\dot{y} = z$, которое одновременно рассматривается как первое уравнение преобразованной задачи. Второе уравнение получается из заданного, проведя в нем замены $\ddot{y} = \dot{z}$ и $\dot{y} = z$

$$m\dot{z} + bz + ry = P(t) \quad \text{или} \quad \dot{z} = \frac{1}{m}[P(t) - bz - ry].$$

Первое начальное условие $y(t_0) = y_0$ полностью подходит новой задаче, а второе $\dot{y}(t_0) = \dot{y}_0$ требует изменения: $z(t_0) = z_0$, где z_0 численно равно \dot{y}_0 .

Окончательно преобразованная задача выглядит следующим образом:

$$\begin{aligned} \dot{y} &= z, & y(t_0) &= y_0, \\ \dot{z} &= \frac{1}{m} [P(t) - bz - ry], & z(t_0) &= z_0 = \dot{y}_0. \end{aligned} \quad (5.9)$$

Если сравнить ее с записью задачи Коши для задачи (5.4, п.5.4), то в данном случае $f(t, y, z) = z$, $g(t, y, z) = \frac{1}{m} [P(t) - bz - ry]$. Теперь ее можно решать методами Рунге-Кутты.

Следует особо отметить, что при численном решении этой задачи (5.9) мы вынуждены находить не только узловые значения y_i , как того требует исходная задача, но и z_i . Это как бы побочный продукт, вызванный нашим неумением численно решать уравнения второго и старшего порядков.

Второй пример – жестко защемленная консоль постоянного сечения, нагруженная произвольной поперечной нагрузкой $q(x)$ (рис. 5.4).

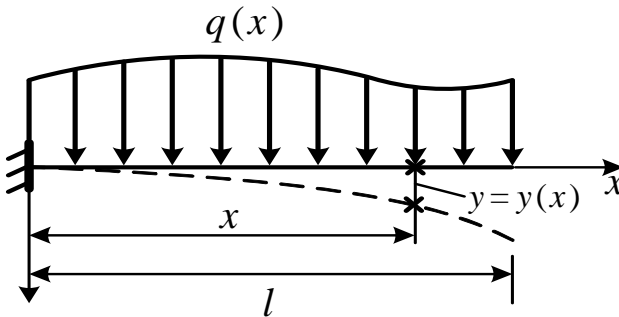


Рис. 5.4. К задаче Коши об изгибе консоли

Задача состоит из обыкновенного дифференциального уравнения четвертого порядка и четырех начальных условий:

$$y^{IV} = \frac{q(x)}{EJ}, \quad (5.10)$$

$$y(0) = 0, y'(0) = 0, M(0) = M_0, Q(0) = Q_0,$$

где $y = y(x)$ – функция прогиба балки;

$EJ = \text{const}$ – жесткость балки на изгиб;

M_0, Q_0 – изгибающий момент и поперечная сила в начальном сечении $x_0 = 0$.

Чтобы преобразовать уравнение четвертого порядка к системе четырех уравнений первого порядка, введем три вспомогательные функции: Θ – угол поворота сечения балки, M – изгибающий момент, Q – поперечную силу, а затем воспользуемся дифференциальными зависимостями, известными из курса сопротивления материалов

$$\Theta = y', M = -EJy'' = -EJ\Theta', Q = M'$$

(здесь учтено, что положительными являются: прогиб – вниз, угол поворота – по часовой стрелке, изгибающий момент – растягивающий нижние волокна, поперечная сила – вращающая отсеченную часть по часовой стрелке).

В итоге получаем следующую преобразованную задачу Коши

$$\begin{aligned} y' &= \Theta, & y(0) &= 0; \\ \Theta' &= -M/EJ, & \Theta(0) &= 0; \\ M' &= Q, & M(0) &= M_0; \\ \Theta' &= -q, & Q(0) &= Q_0. \end{aligned}$$

Эту задачу также можно решить рассмотренными в п. 5.4 методами.

5.7. Решение задачи Коши в системе Mathematica

Численные методы решения дифференциальных уравнений реализуются с помощью функции **NDSolve**:

```
NDSolve[f, y[x], {x, xmin, xmax}]  
NDSolve[{f1, f2, ..., y1(x0), y2(x0), ...}, {y1[x], y2[x],  
...}, {x, xmin, xmax}]
```

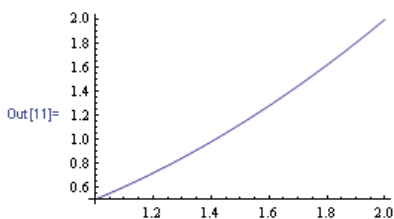
f – дифференциальное уравнение и начальные условия; **f**_{*i*} – *i*-е уравнение системы дифференциальных уравнений; **y**[**x**] – искомая функция, **y**_{*i*}[**x**] – *i*-я искомая функция системы дифференциальных уравнений; **y**_{*i*}(**x**₀) – *i*-е начальное условие; **x**_{min}, **x**_{max} – минимальное и максимальное значения независимой переменной; **x** – аргумент искомой функции.

Рассмотрим решение рассмотренных в лабораторной работе примеров в системе *Mathematica*.

Решение задачи (5.3):

```
▼ In[10]= rez = NDSolve[{y'[t] == 2 * y[t] / t, y[1] == 0.5}, y[t], {t, 1, 2}]  
Plot[y[t] /. rez, {t, 1, 2}]  
rez2 = Table[{t, y[t] /. rez}, {t, 1, 2, 0.2}]  
TableForm[rez2]
```

```
Out[10]= {{y[t] → InterpolatingFunction[{{1., 2.}}, <>] [t]}}
```



```
Out[12]= {{1., {0.5}}, {1.2, {0.72}}, {1.4, {0.98}},  
{1.6, {1.28}}, {1.8, {1.62}}, {2., {2.}}}
```

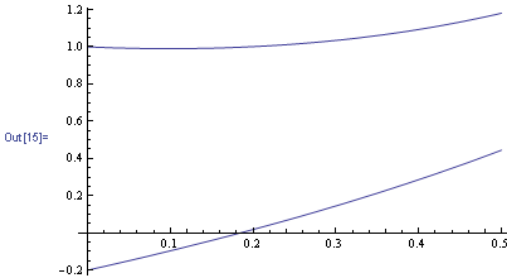
```
Out[13]/TableForm=
```

1.	0.5
1.2	0.72
1.4	0.98
1.6	1.28
1.8	1.62
2.	2.

Решение задачи (5.7):

```
▼ In[14]= rez = NDSolve[{y'[t] == t*y[t] + z[t], z'[t] == t + y[t], y[0] == 1., z[0] == -0.2},  
  {y[t], z[t]}, {t, 0, 0.5}]  
Plot[{y[t], z[t]} /. rez, {t, 0, 0.5}]  
rez2 = Table[{t, {y[t], z[t]} /. rez}, {t, 0, 0.5, 0.1}]  
TableForm[rez2]
```

```
Out[14]= {{y[t] → InterpolatingFunction[{{0., 0.5}}, <=>][t],  
  z[t] → InterpolatingFunction[{{0., 0.5}}, <=>][t]}}
```



```
Out[16]= {{0., {{1., -0.2}}}, {0.1, {{0.9901, -0.0956643}}},  
  {0.2, {{1.00108, 0.018715}}}, {0.3, {{1.03459, 0.145301}}},  
  {0.4, {{1.09325, 0.28647}}}, {0.5, {{1.18079, 0.444913}}}}
```

```
Out[17]/TableForm=  
0.      1.  -0.2  
0.1     0.9901 -0.0956643  
0.2     1.00108 0.018715  
0.3     1.03459 0.145301  
0.4     1.09325 0.28647  
0.5     1.18079 0.444913
```

5.8. Выполнение индивидуальных заданий

Используя трехзначный шифр, выбрать из табл. 5.3 данные к задаче Коши для динамической системы (уравнение 5.8, п. 5.6) с одной степенью свободы. Решить ее методом Рунге-Кутты первого и второго порядка. Результаты представить в виде таблиц и графиков. Решить задание, используя систему Mathematica и сравнить полученные результаты.

Таблица 5.3

Исходные данные к заданию

Шифр №	1 цифра шифра			2 цифра шифра			3 цифра шифра		
	m	b	r	$P(t)$	y_0	y_0	t_0	t_k	Δt
1	2	3	4	5	6	7	8	9	10
1	3,0	0,1	2,4	$\sin(t) + 6,4$	0,1	1,0	0	1	0,1
2	3,5	0,3	2,7	$\sin(2t) - t$	-0,2	-3,0	0,1	2,1	0,2
3	4,0	0,5	3,0	$4\sin(2t)$	0,3	5,0	0,2	3,2	0,3
4	4,5	0,7	3,3	$6\cos(3t)$	-0,4	-7,0	0,3	4,3	0,4
5	5,0	0,9	3,6	$5(t^2 - \cos(t))$	0,5	1,5	0,4	5,4	0,5
6	5,5	0,2	3,9	$6t - \cos(2t)$	-0,6	-2,6	0,5	2,0	0,15
7	6,0	0,4	4,2	$7\sin(3t)$	0,7	8,0	0,6	3,1	0,25
8	6,5	0,6	4,5	$8\cos(2t)$	-0,8	-6,0	0,7	4,2	0,35
9	7,0	0,8	4,8	$0,5t^2 + 7,3$	0,9	4,0	0,8	4,8	0,4
0	7,5	1,0	5,1	$-0,2t + 6,5$	-1,0	2,0	0,9	5,4	0,45

ЛАБОРАТОРНАЯ РАБОТА № 6 РЕШЕНИЕ КРАЕВОЙ ЗАДАЧИ ДЛЯ ЛИНЕЙНЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ МЕТОДОМ КОНЕЧНЫХ РАЗНОСТЕЙ

Цель работы: применить метод конечных разностей к решению краевой задачи для линейных дифференциальных уравнений.

Состав работы:

1. Применение метода конечных разностей к решению линейных дифференциальных уравнений любого порядка;
2. Пример решения линейного дифференциального уравнения с заданными краевыми условиями;
3. Формулировка краевых задач для сжато-изогнутого стержня;
4. Решение краевых задач в системе *Mathematica*;
5. Выполнение индивидуальных заданий.

6.1. Применение метода конечных разностей к решению линейных дифференциальных уравнений любого порядка

Краевыми задачами называются задачи, в которых для решения линейных дифференциального уравнения дополнительные условия

задаются при двух значениях независимой переменной на концах рассматриваемого участка. Конкретные краевые условия (иногда их называют граничные условия) выбираются из физической постановки задачи. Для одного уравнения первого порядка краевую задачу поставить нельзя, так как в этом случае для выделения частного решения требуется только одно дополнительное условие, а в краевой задаче два конца, и, следовательно, минимальное количество дополнительных условий равно двум т. е., краевую задачу можно сформулировать только для дифференциальных уравнений второго и старшего порядков.

Для уравнений второго порядка возможны следующие простейшие варианты краевых условий: 1) на двух концах отрезка задаются значения искомой функции; 2) на одном из концов отрезка задается значение искомой функции, на втором – значение первой производной.

Основная идея метода конечных разностей (МКР) состоит в том, что он сводит решение краевой задачи к решению системы алгебраических уравнений относительно значений искомой функции (решения) на заданном множестве точек. Это достигается заменой производных, входящих в дифференциальное уравнение и в краевые условия, их **конечными разностями**.

Рассмотрим аппроксимацию первой производной для функции $y = f(x)$, заданной таблично в виде $y = y_1, y_2, \dots, y_{n-1}, y_n$ при $x = x_1, x_2, \dots, x_{n-1}, x_n$ (рис. 6.1). Считаем, что узлы равноудалены друг от друга. Тогда шаг Δx (разность между соседними значениями аргумента) будет постоянным.

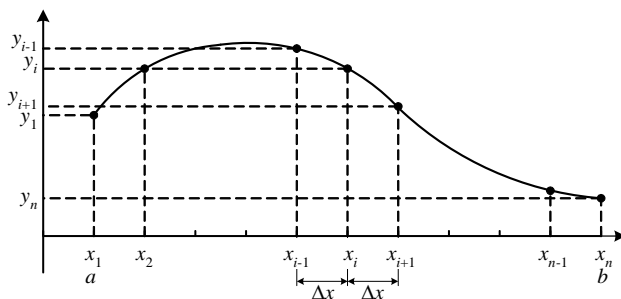


Рис. 6.1. Фрагмент отрезка $[a, b]$ с нанесенными на нем узлами и узловыми ординатами искомой функции

В зависимости от способа вычисления конечных разностей получают разные формулы для вычисления первой производной (первая разность) в одной и той же точке:

$$\text{левая разность } y'_i \approx \frac{y_i - y_{i-1}}{\Delta x}; \quad (6.1)$$

$$\text{правая разность } y'_i \approx \frac{y_{i+1} - y_i}{\Delta x}; \quad (6.2)$$

$$\text{центральная разность } y'_i \approx \frac{y_{i+1} - y_{i-1}}{2\Delta x}. \quad (6.3)$$

Конечно-разностные выражения для старших производных получаются по принципу «разность от разности». Можно получить формулы для

– второй производной (вторая разность)

$$y''_i = (y'_i)' \approx \frac{y'_i - y'_{i-1}}{\Delta x} = \frac{(y_{i+1} - y_i) / \Delta x - (y_i - y_{i-1}) / \Delta x}{\Delta x} = \frac{y_{i-1} - 2y_i + y_{i+1}}{\Delta x^2}; \quad (6.4)$$

– третьей производной (третья разность)

$$\begin{aligned} y'''_i &\approx \frac{y''_{i+1} - y''_{i-1}}{2\Delta x} = \frac{(y_i - 2y_{i+1} + y_{i+2}) / \Delta x^2 - (y_{i-2} - 2y_{i-1} + y_i) / \Delta x^2}{2\Delta x} = \\ &= \frac{-y_{i-2} + 2y_{i-1} - 2y_{i+1} + y_{i+2}}{2\Delta x^3}; \end{aligned} \quad (6.5)$$

– четвертой производной (четвертая разность)

$$\begin{aligned} y^{IV}_i &= \frac{y''_{i-1} - 2y''_i + y''_{i+1}}{\Delta x^2} = \frac{(y_{i-2} - 2y_{i-1} + y_i) - 2(y_{i-1} - 2y_i + y_{i+1}) + (y_i - 2y_{i+1} + y_{i+2})}{\Delta x^4} = \\ &= \frac{y_{i-2} - 4y_{i-1} + 6y_i - 4y_{i+1} + y_{i+2}}{\Delta x^4} \end{aligned} \quad (6.6)$$

и т. д., являющиеся формулами численного дифференцирования, точность которых существенно зависит от Δx .

Пусть имеется краевая задача, состоящая из линейного дифференциального уравнения второго или старших порядков и краевых условий на концах отрезка $[a, b]$. Требуется найти значения искомой функции в узлах, равномерно расположенных на отрезке. Решение краевой задачи методом конечных разностей можно разделить на четыре основных этапа.

Этап 1. Подготовка к решению задачи.

Отрезок $[a, b]$ разбивается на m частей (шагов) длиной $\Delta x = (b - a) / m$. Точки деления (узлы) нумеруются слева направо: например $1, 2, \dots, m$. Узлы 1 и m называются контурными, а узлы $2, 3, \dots, m-2, m-1$ – внутренними. В некоторых случаях приходится вводить еще и законтурные узлы, расположенные за пределами отрезка. Неизвестными задачи выступают узловые значения искомой функции во внутренних узлах. Если значение искомой функции в одном из контурных узлов не задано краевыми условиями, то в число неизвестных надо включить узловое значение функции в соответствующем контурном узле.

Этап 2. Составление прямоугольной СЛАУ.

В решаемом дифференциальном уравнении аргумент x заменяется узловым значением x_i , искомая функция $y(x)$ – узловым значением $y_i = y(x_i)$, производные искомой функции $y'(x), y''(x), \dots$ – соответствующими конечными разностями y'_i, y''_i, \dots . Если в уравнении содержится некоторая неизвестная функция аргумента x , например, $f(x)$, то она также заменяется узловым значением $f_i = f(x_i)$.

В результате таких замен дифференциальное уравнение превращается в алгебраическое, называемое конечно-разностным для произвольного узла i или шаблоном.

По этому шаблону составляются уравнения :во-первых, во всех внутренних узлах ($i = 2, \dots, m-1$), во-вторых, для того контурного узла 1 или m , в котором не задана величина y_a или y_b .

В совокупности составленные уравнения образуют СЛАУ, имеющую прямоугольную структуру, т.к. число неизвестных в ней превышает количество уравнений. Причем превышает на величину, равную порядку решаемого уравнения или, что одно и то же, количеству краевых условий.

Этап 3. Приведение прямоугольной СЛАУ к квадратному виду.

Такое приведение может быть выполнено, по крайней мере,

двумя способами. Первый состоит в том, что к составленным добавляют недостающие уравнения, в качестве которых используются краевые условия. Второй способ заключается в преобразовании составленных уравнений с помощью краевых условий, в результате чего число неизвестных сокращается до числа уравнений. Первый способ более прост, как при ручном, так и автоматизированном формировании системы, но ее порядок оказывается выше, чем при использовании второго способа.

Этап 4. Решение квадратной СЛАУ.

Используя любой метод (например, метод Гаусса, метод прогонки и т. д.) систему решают и получают приближенные узловые значения искомой функции. Приближенность результатов решения краевой задачи методом конечных разностей обусловлена влиянием ошибки метода и ошибки округления. Первая возникает вследствие замены производных приближенными конечно-разностными выражениями, а вторая – главным образом при решении СЛАУ. Сделать первую ошибку малой можно, дробя шаг (при этом возрастают размеры СЛАУ), а вторую – переходя к машинной арифметике удвоенной точности.

6.2. Пример решения линейного дифференциального уравнения с заданными краевыми условиями методом конечных разностей

Настоящий пример не ставит своей целью получение результатов высокой точности, а служит для понимания МКР. Поэтому шаг Δx в задаче взят довольно крупным.

Задача. Решить методом конечных разностей краевую задачу (число шагов $m = 4$)

$$x^2 y'' - 2xy' + 2y = x^3;$$

$$y(1) = 1,1;$$

$$y(2) = -0,5.$$

Решение.

1. Отрезок $[a, b]$, где $a = 1$; $b = 2$, разбиваем на $m = 4$ частей (шагов) длиной $\Delta x = (2 - 1)/4 = 0,25$.

Узлы нумеруем слева направо от 1 до 5. Неизвестными являются величины y_2, y_3, y_4 (рис. 6.2). Известные величины – $y_1 = y_a = 1,1$ (при $x_1 = 1$) и $y_5 = y_b = -0,5$ (при $x_5 = 2$).

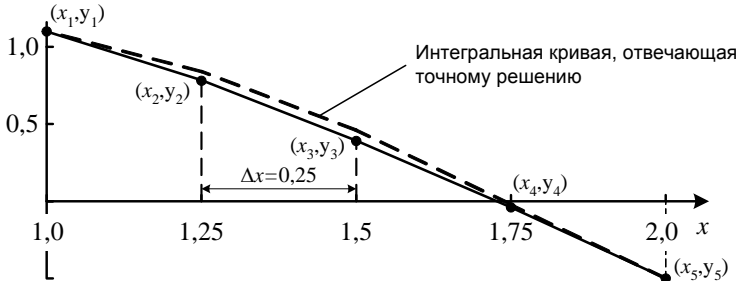


Рис. 6.2. К решению краевой задачи

2. Используя конечно-разностные выражения (6.3) и (6.4), преобразуем исходное дифференциальное уравнение

$$x_i^2 \frac{y_{i-1} - 2y_i + y_{i+1}}{\Delta x^2} - 2x_i \frac{y_{i+1} - y_{i-1}}{2\Delta x} + 2y_i = x_i^3.$$

Умножая все уравнения на Δx^2 и приводя подобные слагаемые, получим

$$(x_i^2 + x_i \Delta x) y_{i-1} + 2(\Delta x^2 - x_i^2) y_i + (x_i^2 - x_i \Delta x) y_{i+1} = x_i^3 \Delta x^2.$$

Учитывая, что $\Delta x = 0,25$; $x_2 = 1,25$; $x_3 = 1,5$; $x_4 = 1,75$, составляем систему трех алгебраических уравнений

$$i = 2:$$

$$(1,25^2 + 1,25 \cdot 0,25) y_1 + 2(0,25^2 - 1,25^2) y_2 + (1,25^2 - 1,25 \cdot 0,25) y_3 = 1,25^3 \cdot 0,25^2;$$

$$i = 3:$$

$$(1,5^2 + 1,5 \cdot 0,25) y_2 + 2(0,25^2 - 1,5^2) y_3 + (1,5^2 - 1,5 \cdot 0,25) y_4 = 1,5^3 \cdot 0,25^2;$$

$$i = 4:$$

$$(1,75^2 + 1,75 \cdot 0,25) y_3 + 2(0,25^2 - 1,75^2) y_4 + (1,75^2 - 1,75 \cdot 0,25) y_5 = 1,75^3 \cdot 0,25^2.$$

Подсчитывая значения коэффициентов, записываем СЛАУ.

$$\begin{array}{rclcl} 1,875y_1 & -3y_2 & +0,9375y_3 & = & 0,122070, \\ 2,625y_2 & -4,375y_3 & +1,875y_4 & = & 0,210938, \\ 3,5y_3 & -6y_4 & +2,625y_5 & = & 0,334961. \end{array}$$

Получили систему трех уравнений с пятью узловыми значениями искомой функции: y_1, y_2, y_3, y_4, y_5 .

3. Для того, чтобы избавиться от «лишних» переменных, воспользуемся вторым способом, описанном в этапе 3 п. 6.1.

Вместо y_1 в первом уравнении подставляем значение $y_a = 1,1$, а вместо y_5 в третьем уравнении – значение $y_b = -0,5$. Переносим числа $1,875 \cdot 1,1$ и $2,625 \cdot (-0,5)$ в правую часть соответственно первого и третьего уравнений и алгебраически складывая их со стоящими там свободными членами, окончательно получим квадратную СЛАУ третьего порядка и ленточной (трехдиагональной) структуры:

$$\begin{array}{rclcl} -3y_2 & +0,9375y_3 & & = & -1,94043, \\ 2,625y_2 & -4,375y_3 & +1,875y_4 & = & 0,210938, \\ & 3,5y_3 & -6y_4 & = & 1,64746. \end{array} \quad (6.7)$$

4. Решая полученную систему линейных уравнений любым из известных методов, находим

$$y_2 = 0,770252, y_3 = 0,395015, y_4 = -0,0441510.$$

Тогда исходная задача имеет следующее решение

$$y = (1,1; 0,770252; 0,395015; -0,0441510; -0,5) \\ \text{при } x = (1; 1,25; 1,5; 1,75; 2).$$

По этим значениям на рис. 6.2 построена сплошная линия.

Рассмотренная краевая задача имеет точное аналитическое решение

$$y(x) = 0,5x(6,9 - 5,7x + x^2),$$

которому отвечает пунктирная линия на рисунке 6.2.

При этом

$$y(x_2) = 0,835938, \quad y(x_3) = 0,45, \quad y(x_4) = -0,0109375.$$

Как и ожидалось, метод конечных разностей при столь большом шаге дает только качественно правильное решение.

6.3. Формулировка краевых задач для сжато-изогнутого стержня

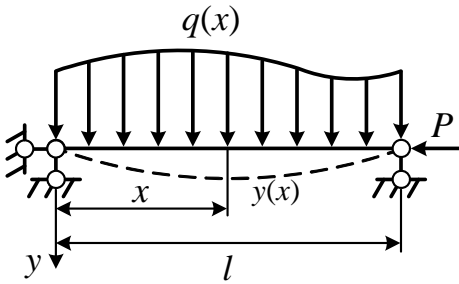
Рассмотрим упругий стержень длиной l постоянного поперечного сечения с жесткостью на изгиб $EJ = const$, нагруженный произвольной поперечной нагрузкой $q(x)$ и сжатый силой P , не превышающей своего критического значения P_{cr} . Как известно из курса сопротивления материалов, работа такого стержня описывается обыкновенным дифференциальным уравнением четвертого порядка

$$EJy^{IV} + Py'' = q(x), \quad (6.8)$$

где $y = y(x)$ – функция прогибов оси стержня, и четырьмя краевыми условиями, отражающими характер опирания стержня по двум его концам.

Приведем формулировку краевых условий и формулы критических сил для шести схем стержня. Во всех схемах начало координат совмещено с левым концом стержня. Ось x направлена слева направо вдоль оси стержня, так что $a = 0$, $b = l$. Ось прогибов y направлена вниз, такое же направление имеет и нагрузка $q(x)$.

Схема 1

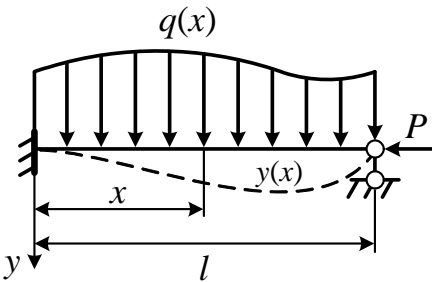


$$y(0) = 0, y(l) = 0, \\ y''(0) = 0, y''(l) = 0$$

(это значит, что на шарнирной опоре равны нулю прогиб и изгибающий момент (см. также примечание после схемы б)).

$$P_{cr} = \pi^2 \frac{EJ}{l^2}.$$

Схема 2

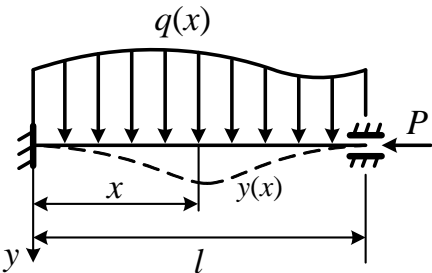


$$y(0) = 0, y(l) = 0, \\ y'(0) = 0, y''(l) = 0$$

(в защемлении равны нулю прогиб и угол поворота).

$$P_{cr} = \pi^2 \frac{EJ}{(0,7l)^2}.$$

Схема 3

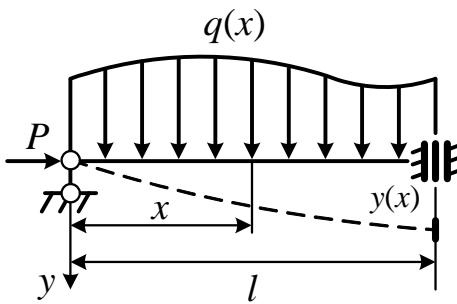


$$y(0) = 0, y(l) = 0, \\ y'(0) = 0, y'(l) = 0$$

(скользящее защемление правого конца стержня при малых прогибах описывается так же, как и обычная заделка).

$$P_{cr} = \pi^2 \frac{EJ}{(0,5l)^2}.$$

Схема 4



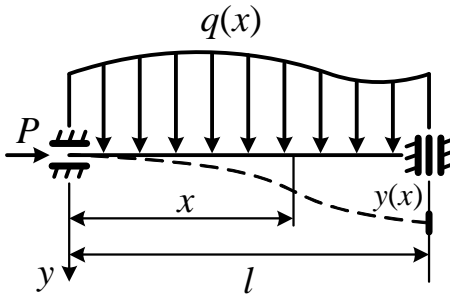
$$y(0) = 0, y'(l) = 0,$$

$$y''(0) = 0, y'''(l) = 0$$

(в скользящем защемлении на правом конце равны нулю угол поворота и поперечная сила).

$$P_{cr} = \pi^2 \frac{EJ}{(2l)^2}.$$

Схема 5

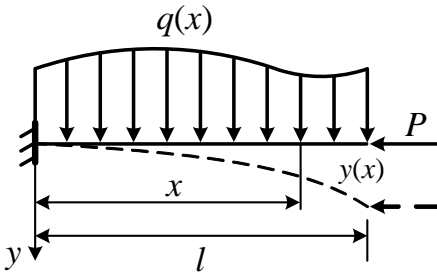


$$y(0) = 0, y'(l) = 0,$$

$$y'(0) = 0, y'''(l) = 0,$$

$$P_{cr} = \pi^2 \frac{EJ}{l^2}.$$

Схема 6



$$y(0) = 0, y''(l) = 0,$$

$$y'(0) = 0, -EJy'''(l) = Py'(l),$$

$$P_{cr} = \pi^2 \frac{EJ}{(2l)^2}.$$

Примечание.

При формулировке краевых условий учтены дифференциальные зависимости при изгибе, известные из курса сопротивления материалов:

$$y' = \Theta, M = -EJy'', Q = -EJy''''.$$

Отсюда равенство нулю угла поворота Θ в заделке $x = 0, x = l$ записывается как $y'(0) = 0$ или $y'(l) = 0$. Равенство нулю изгибающего момента в шарнире и на концах $x = 0, x = l$ отражается в формулировке $y''(0) = 0$ или $y''(l) = 0$.

Аналогично записывается четвертое краевое условие в схемах 4 и 5:

$$y'''(l) = 0.$$

Оно отражает факт равенства нулю поперечной силы Q на правом конце балки $x = l$.

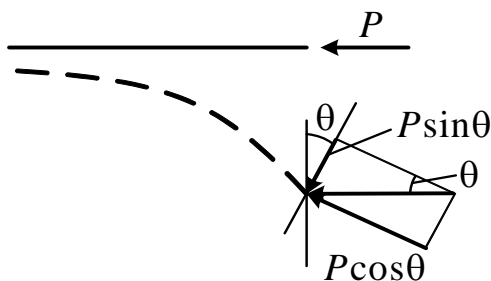


Рис. 6.3. К формулировке четвертого краевого условия в схеме б

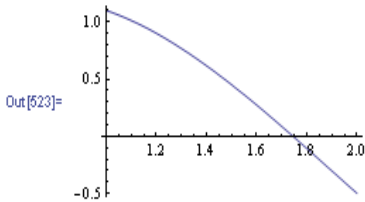
Несколько сложнее формулируется краевое условие в схеме б. При изгибе стержня его сечение поворачивается на угол Θ (рис. 6.3), а сжимающая сила P сохраняет свое прежнее (горизонтальное) положение. Разложив ее на два направления (нормальное и касательное к оси деформированного стержня), получаем, что на конце стержня $N = -P \cos\Theta \approx -P, Q = P \sin\Theta \approx P y'$. Замены $\cos\Theta \approx 1, \sin\Theta \approx 0$ возможны только при малых прогибах.

6.4. Решение краевых задач в системе Mathematica

Для решения линейного дифференциального уравнения с заданными краевыми условиями из п. 6.2 воспользуемся функцией системы Mathematica **NDSolve** (см. ее описание в лабораторной работе № 5)

```
In[522]:= rez = NDSolve[{x^2 * y''[x] - 2 * x * y'[x] + 2 * y[x] == x^3,  
    y[1.] == 1.1, y[2.] == -0.5}, y[x], {x, 1., 2.}]  
Plot[y[x] /. rez, {x, 1., 2.}]  
rez2 = Table[{x, y[x] /. rez}, {x, 1., 2., 0.25}]  
TableForm[rez2]
```

```
Out[522]= {{y[x] → InterpolatingFunction[{{1., 2.}}, <=>][x]}}
```



```
Out[524]= {{1., {1.1}}, {1.25, {0.835938}},  
    {1.5, {0.45}}, {1.75, {-0.0109375}}, {2., {-0.5}}}
```

```
Out[525]/TableForm=
```

1.	1.1
1.25	0.835938
1.5	0.45
1.75	-0.0109375
2.	-0.5

6.5. Выполнение индивидуальных заданий

Задана краевая задача, состоящая из обыкновенного дифференциального уравнения второго порядка

$$y'' + ky = f(x)$$

и двух краевых условий $y(0) = y(l) = 0$. Значения $k = k^*(\pi/l)^2$, l и выражение $f(x)$ принимаются по табл. 6.3. Используя метод

конечных разностей при $m = 4$, составить таблицу с результатами и построить график функции $y = y(x)$. При решении системы линейных алгебраических уравнений использовать *метод прогонки*. Решить задание, используя систему *Mathematica* и сравнить полученные результаты.

Таблица 6.3

Исходные данные к заданию

Первая цифра шифра	k^*	Вторая цифра шифра	l	Третья цифра шифра	$f(x)$
1	2	3	4	5	6
0	0,65	0	7,6	0	$\frac{0,01}{l}(x^2 - lx)$
1	0,45	1	5,6	1	$\frac{-0,15}{l^2}x \sin \frac{\pi x}{l}$
2	0,5	2	8,4	2	$\frac{8(l-3x)}{l^3(x+1)}$
3	0,55	3	6,4	3	$\frac{x^3 - l^2x}{25l^2}$
4	0,8	4	6,8	4	$-\frac{x}{5l^2} \ln \left(\frac{2x}{l} + 1 \right)$
5	0,6	5	7,2	5	$\frac{-1,5}{l^3} \sqrt{\frac{x}{l} + 2}$
6	0,35	6	6,0	6	$-1,5 \frac{\sqrt{x}}{l^3} \sin \frac{\pi x}{l}$
7	0,75	7	8,8	7	$\frac{2lx^2 - x^3 - l^3}{100l^2}$
8	0,7	8	8,0	8	$\frac{15 \ln(x^2 + 2)}{l^3(x-2l)}$
9	0,4	9	5,2	9	$\frac{3}{l^3} \left(\frac{4x}{l} - 3 \right)$

Указания к работе.

1. Работа шарнирно опертого по обоим концам стержня длиной l постоянной жесткости на изгиб EJ , нагруженного произвольной поперечной нагрузкой и сжатой силой P (рис. 6.4), описывается обыкновенным дифференциальным уравнением

$$y'' + \frac{P}{EJ} y = \frac{M(x)}{EJ}$$

и краевыми условиями $y(0) = y(l) = 0$, где $y = y(x)$ – функция прогибов стержня; $M(x)$ – изгибающий момент в произвольном сечении x .

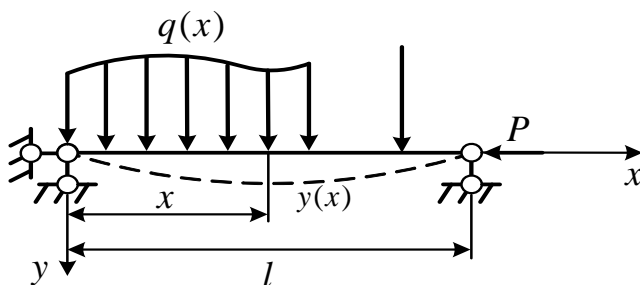


Рис. 6.4. Расчетная схема сжато-изогнутого стержня

Введя обозначения $k = P/EJ$ и $f(x) = M(x)/EJ$, получаем заданное уравнение.

2. Разделив пролет стержня на 4 части ($m = 4$), надо отметить на оси x пять узлов 1, 2, ..., 5, отстоящих друг от друга на шаг $\Delta x = l/4$. При составлении конечно-разностного шаблона решаемого уравнения правая часть заменяется узловым значением $f_i = f(x_i)$, последовательно полагая i равным 2, 3, 4, получают три уравнения с пятью неизвестными: y_1, y_2, y_3, y_4, y_5 . Чтобы сделать из этих уравнений квадратную СЛАУ, надо положить в них (в соответствии с краевыми условиями) $y_1 = y_5 = 0$. Решив эту систему, находят приближенные значения прогибов стержня. Выбрав масштаб по своему усмотрению, нужно построить график прогибов стержня (положительные значения прогиба откладываются вниз от оси).

3. В лабораторной работе № 2 для решения систем линейных алгебраических уравнений использовался метод Гаусса. Вместе с тем, для решения СЛАУ специального вида, например, с трехдиагональной структурой (6.7), можно использовать **метод прогонки**.

Рассмотрим СЛАУ 3 ($n = 3$) порядка следующего вида:

$$\begin{aligned} b_1 x_1 + c_1 x_2 &= d_1, \\ a_2 x_1 + b_2 x_2 + c_2 x_3 &= d_2, \\ a_3 x_2 + b_3 x_3 &= d_3 \end{aligned}$$

Метод состоит из двух ходов и напоминает метод Гаусса.

Прямой ход заключается в поиске коэффициентов P и Q по следующим формулам:

$$\begin{aligned} P_1 &= 0, & Q_1 &= 0, \\ P_{i+1} &= \frac{-c_i}{a_i P_i + b_i}, & Q_{i+1} &= \frac{d_i - a_i Q_i}{a_i P_i + b_i}, \quad i = \overline{1, n}. \end{aligned}$$

Во время **обратного хода** непосредственно определяются значения неизвестных в обратной последовательности:

$$\begin{aligned} x_n &= Q_{n+1}, \\ x_i &= Q_{i+1} + P_{i+1} x_{i+1}, \quad i = \overline{n-1, n-2, \dots, 1}. \end{aligned}$$

ЛАБОРАТОРНАЯ РАБОТА № 7
ЧИСЛЕННЫЕ МЕТОДЫ ОПТИМИЗАЦИИ.
ГРАФИЧЕСКИЙ МЕТОД РЕШЕНИЯ ЗАДАЧ ЛИНЕЙНОГО
ПРОГРАММИРОВАНИЯ

Цель работы: изучить графический метод решения задач линейного программирования.

Состав работы:

1. Общие сведения о задачах оптимизации;
2. Графическое решение ЗЛП;
3. Решение задач линейного программирования в системе *Mathematica*;
4. Выполнение индивидуальных заданий.

7.1. Общие сведения о задачах оптимизации

В настоящей работе будем рассматривать задачи оптимизации, имеющие следующую структуру:

- 1) задана функция цели

$$Z = f(x_1, x_2, \dots, x_n),$$

которая принимает максимальное или минимальное значение;

- 2) присутствуют m ограничений

$$g_i(x_1, x_2, \dots, x_n) \left\{ \begin{array}{l} \leq \\ \geq \\ = \end{array} \right\} b_i, \quad i = 1, \dots, m;$$

- 3) задано условия неотрицательности переменных

$$x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0.$$

Задача оптимизации в такой постановке называется условной. Вторая и третья части могут отсутствовать, тогда задача называется безусловной.

Решить задачу оптимизации – это значит найти такие значения переменных x_1, x_2, \dots, x_n , которые удовлетворяют ограничениям, условиям неотрицательности и сообщают функции цели максимальное или минимальное значение.

Совокупность указанных значений переменных x_1, x_2, \dots, x_n будем называть **оптимальным** планом.

Задачи оптимизации описывают явления и процессы в различных областях науки, техники, экономики. Большую роль они играют и в строительстве (оптимальное проектирование конструкций, транспортная задача и другие.)

Пусть требуется найти оптимальные размеры b, h прямоугольного сечения простой балки пролетом l , загруженной равномерно распределенной нагрузкой интенсивности q (рис. 7.1). Модуль упругости материала E , расчетное сопротивление на растяжение-сжатие R . Допустимый прогиб балки равен s . В качестве функции цели возьмем объем материала, а в качестве ограничений – условия прочности и жесткости.

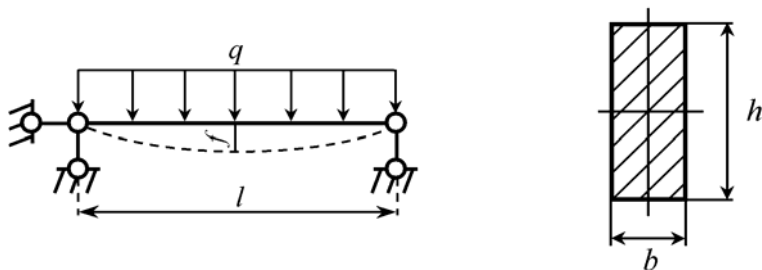


Рис. 7.1. К формулировке задачи оптимизации однопролетной балки

Объем материала балки $V = bhl$ должен быть минимальным. Но поскольку пролет l – известная (заданная) величина, то в качестве функции цели можно принять площадь сечения. И тогда, вводя обозначения $V = Z, b = x_1, h = x_2$, получаем

$$Z = x_1 x_2 \rightarrow \min.$$

Наибольшие нормальные напряжения возникают в фибровых волокнах на середине балки, где действует наибольший изгибающий момент. Причем эти напряжения

$$\max \sigma = \frac{\max M}{W}$$

не должны превышать расчетного сопротивления R . Учитывая, что $\max M = ql^2/8$, а момент сопротивления сечения $W = bh^2/6$, получаем

$$\frac{ql^2}{8} \cdot \frac{6}{bh^2} \leq R,$$

что после преобразований и замены $b = x_1$, $h = x_2$ дает первое ограничение

$$x_1 x_2^2 \geq \frac{3ql^2}{4R}.$$

Наибольший прогиб в балке возникает также в середине пролета

$$f = \frac{5}{384} \frac{ql^4}{EJ}.$$

Он не должен превышать допустимого прогиба c , т. е.

$$\frac{5}{384} \frac{ql^4}{EJ} \leq c.$$

Момент инерции прямоугольного сечения $J = bh^3/12$. С учетом этого выражения второе ограничение примет вид

$$x_1 x_2^3 \geq \frac{5ql}{32Ec}.$$

Формулируя задачу оптимизации, нужно также учесть, что искомые величины b и h не должны принимать отрицательных значений. В итоге получаем задачу оптимизации

$$Z = x_1 x_2 \rightarrow \min,$$

$$x_1 x_2^2 \geq \frac{3ql^2}{4R},$$

$$x_1 x_2^3 \geq \frac{5ql}{32Ec},$$

$$x_1 \geq 0, x_2 \geq 0.$$

Раздел математики, изучающий задачи оптимизации и разрабатывающий методы их решения, называется **математическим программированием** (иногда в литературе встречается термин «математическое планирование»).

В зависимости от вида функции цели и ограничений математическое программирование делится на линейное, нелинейное, целочисленное, стохастическое, динамическое и т. д.

В задачах **линейного программирования** (ЗЛП) функции цели и ограничения линейны относительно x_1, x_2, \dots, x_n :

$$Z = c_1 x_1 + c_2 x_2 + \dots + c_n x_n \rightarrow \max \text{ (или min),}$$

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \geq b_1,$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \geq b_2,$$

.....

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \geq b_m,$$

$$x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0.$$

В рассмотренном выше примере была сформулирована задача **нелинейного программирования** (ЗНП) – в этих задач функция цели или хотя бы одно из ограничений нелинейны относительно переменных. Эти задачи настолько сложны, что до сих пор не удалось разработать общие методы (подобно симплекс-методу в линейном планировании), которые гарантировали получение оптимального решения любой ЗНП. Имеются методы решения лишь отдельных специальных классов задач и, прежде всего, задач с выпуклой функцией цели и выпуклыми ограничениями.

В задачах **целочисленного программирования** неизвестные могут принимать только целочисленные значения. Если в целевой функции или в функциях, определяющих область возможных изме-

нений переменных, содержатся случайные величины, то такая задача относится к задаче *стохастического программирования*. Задача, процесс нахождения решения которой является многоэтапным, относится к задаче *динамического программирования*.

В настоящем практикуме рассматриваются ЗЛП и два метода их решения – графический и симплекс-метод.

7.2. Графическое решение ЗЛП

Решение таким методом выполняется для ЗЛП с двумя неизвестными.

Рассмотрим алгоритм графического решения на конкретном примере

$$Z = x_1 - 2x_2 \rightarrow \begin{matrix} \min \\ \max \end{matrix}; \quad \text{целевая функция}$$

$$\left. \begin{array}{l} 3x_1 + 4x_2 \leq 12; \\ 2x_1 - x_2 \geq -2; \\ x_1 - 2x_2 \leq 6; \\ x_1 \geq 0, \end{array} \right\} \begin{array}{l} \text{линейные ограничения} \\ \text{условие неотрицательности} \end{array}$$

где условие неотрицательности наложено только на первую неизвестную. По существу, здесь две задачи: одна – это задача минимизации Z , а вторая – максимизации Z .

Решение состоит из четырех этапов.

Этап 1. Построение области допустимых решений.

Областью допустимых решений (ОДР) в задаче с двумя неизвестными называется пересечение полуплоскостей, определяемых ограничениями и условиями неотрицательности. Иначе говоря, это область на плоскости x_1Ox_2 , состоящая из точек, координаты которых удовлетворяют как всем ограничениям, так и условиям неотрицательности.

Первое ограничение $3x_1 + 4x_2 \leq 12$ определяет первую полуплоскость. Чтобы найти ее, нужно сначала построить граничную прямую $3x_1 + 4x_2 = 12$. В данном случае она проходит через точки $(0,3)$ и $(4,0)$ (см. рис. 7.2). Влево и вниз от нее идет одна бесконечная полуплоскость, а вправо и вверх – вторая полуплоскость. Какая

из них отвечает ограничению $3x_1 + 4x_2 \leq 12$? Для ответа на этот вопрос возьмем какую-нибудь пробную точку. Если граничная прямая не проходит через начало координат, то в качестве пробной точки проще всего использовать именно начало координат, т. е. точку $(0,0)$. Подставляя эти координаты в левую часть первого ограничения, получаем $0 < 12$. Таким образом, искомая полуплоскость включает начало координат. Направление полуплоскости показываем стрелками, рядом с которыми пишем номер ограничения.

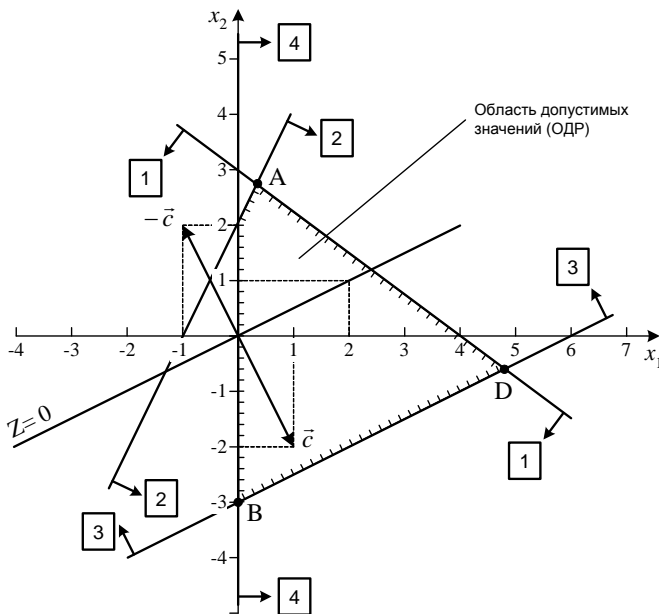


Рис. 7.2. Графическое решение ЗЛП с двумя неизвестными

ограничения, получаем $0 < 12$. Таким образом, искомая полуплоскость включает начало координат. Направление полуплоскости показываем стрелками, рядом с которыми пишем номер ограничения.

Аналогично строятся полуплоскости 2 и 3. Условие неотрицательности $x_1 \geq 0$ дает четвертую полуплоскость.

Фигура, очерченная четырьмя граничными прямыми, и есть ОДР. Она представляет собой замкнутую выпуклую область.

Этап 2. Построение линии уровня $Z = 0$.

Линией уровня некоторой функции называется такая линия, во всех точках которой значение этой функции одинаково.

Линия уровня функции цели при $Z = 0$ – это прямая $c_1x_1 + c_2x_2 = 0$ или, в нашем случае, $x_1 - 2x_2 = 0$. Ее график проходит через начало координат $(0, 0)$ и, например, точку с координатами $(2, 1)$. Во всех точках этой линии $Z = 0$.

Этап 3. Построение вектора-градиента (или вектора-антиградиента).

Вектором-градиентом некоторой функции нескольких переменных называется вектор, указывающий направление наискорейшего возрастания функции. Компоненты вектора-градиента равны частным производным от функции по всем ее переменным. В ЗЛП с двумя переменными $Z = c_1x_1 + c_2x_2$ и

$$\vec{c} = \begin{bmatrix} \partial Z / \partial x_1 \\ \partial Z / \partial x_2 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}.$$

В рассматриваемой задаче $\vec{c} = [1, -2]^T$.

Соответственно **вектор-антиградиент** указывает направление наискорейшего убывания функции цели

$$-\vec{c} = \begin{bmatrix} -\partial Z / \partial x_1 \\ -\partial Z / \partial x_2 \end{bmatrix} = \begin{bmatrix} -c_1 \\ -c_2 \end{bmatrix}.$$

В нашей задаче $-\vec{c} = [-1, 2]^T$.

Чтобы построить \vec{c} , от начала координат откладываем вдоль x_1 первую координату (равную +1), а вдоль x_2 – вторую (равную -2). На этих отрезках строим прямоугольник, диагональ которого, снабженная направлением, и есть искомый вектор-градиент. Он перпендикулярен линии уровня функции цели. Аналогично строится $-\vec{c}$, который направлен в сторону, противоположную \vec{c} .

Этап 4А. Решение задачи минимизации Z .

Если линию уровня перемещать параллельно самой себе в направлении вектора-антиградиента, то функция цели будет убывать, принимая отрицательные значения все большего модуля.

И это будет продолжаться до тех пор, пока линия уровня не коснется крайней точки ОДР, а именно, вершины A . Координаты этой вершины и являются искомым решением задачи минимизации Z . Для их определения надо решить систему двух уравнений, соответствующих первой и второй граничным прямым

$$3x_1 + 4x_2 = 12$$

$$2x_1 - x_2 = -2,$$

откуда $x_1 = 4/11 = 0,363636$, $x_2 = 30/11 = 2,72727$ (эти значения называются оптимальным планом при решении задачи минимизации Z).

Функция цели принимает в оптимальной вершине A значение $\min Z = 4/11 - 2 \cdot 30/11 = -56/11 = -5,09091$.

Этап 4Б. Решение задачи максимизации Z .

Чтобы найти оптимальный план в задаче максимизации Z , линию уровня надо перемещать в направлении вектора-градиента. При этом функция цели будет возрастать. Перемещение линии уровня надо проводить до того момента, пока она не коснется крайней точки ОДР.

Но данная задача имеет особенность: линия уровня параллельна одной из сторон ОДР, а именно стороне BD . Поэтому при поступательном движении линия уровня коснется границы ОДР в бесконечно большем количестве точек – в точках B , D и во всех промежуточных точках. Отсюда следует, что задача максимизации Z в данном случае имеет бесконечное множество решений: оптимальным планом являются координаты любой точки отрезка BD .

Возьмем для определенности в качестве оптимальной вершины точку B . Ее координаты (оптимальный план) видны на чертеже

$$x_1 = 0, x_2 = -3, \max Z = 0 - 2 \cdot (-3) = 6.$$

При решении некоторой ЗЛП может оказаться, что одни ограничения противоречат другим (рис. 7.3). Так что найти их пересечение не представляется возможным. В этом случае говорят, что ОДР пустая. Такая задача решения не имеет.

Встречаются случаи, когда в направлении вектора-градиента или антиградиента ОДР не замкнута (рис. 7.4). Двигаясь в этом направлении, достичь крайней точки ОДР невозможно. Такая ЗЛП также не имеет решения.

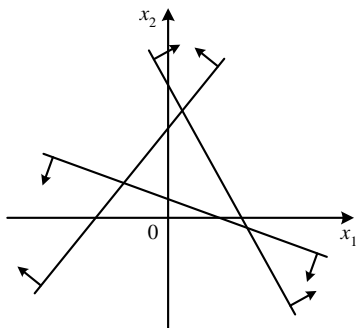


Рис. 7.3. ОДР пустая (ограничения противоречивы)

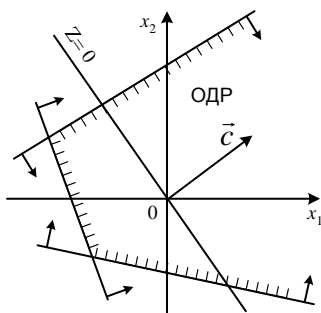


Рис. 7.4. Задача максимизации Z решения не имеет

7.3. Решение задач линейного программирования в системе Mathematica

Для поиска максимума и минимума аналитически заданной функции нескольких аргументов при наличии ограничений с помощью системы *Mathematica* можно использовать следующие функции:

NMaximize [{ **f**, **cons** }, { **x**, **y**, ... }] ищет максимум целевой функции **f** в области, определяемой списком ограничений (неравенствами) **cons**, { **x**, **y**, ... } – список переменных;

Nminimize [{ **f**, **cons** }, { **x**, **y**, ... }] ищет минимум целевой функции **f** в области, определяемой списком ограничений (неравенствами) **cons**, { **x**, **y**, ... } – список переменных.

Решим рассматриваемый в лабораторной работе пример с помощью системы *Mathematica*

```
In[182]= NMinimize [ {x1 - 2 * x2, 3 * x1 + 4 * x2 ≤ 12,
                    2 * x1 - x2 >= -2, x1 - 2 * x2 ≤ 6}, {x1, x2} ]
Out[182]= {-5.09091, {x1 → 0.363636, x2 → 2.72727}}
```

```
In[183]= NMaximize [ {x1 - 2 * x2, 3 * x1 + 4 * x2 ≤ 12,
                    2 * x1 - x2 >= -2, x1 - 2 * x2 ≤ 6}, {x1, x2} ]
Out[183]= {6., {x1 → 0., x2 → -3.}}
```

Результаты, полученные в системе *Mathematica* и с помощью графического метода, совпадают.

7.4. Выполнение индивидуальных заданий

Решить графически задачу линейного программирования с двумя неизвестными. Исходные данные выбрать по табл. 7.1 в соответствии с шифром. Проверить полученные результаты, используя систему *Mathematica*.

Указания к работе. Функция цели в задаче может достигать как максимума, так и минимума. Таким образом, надо решать по существу две ЗЛП. Условия неотрицательности на переменные x_1 и x_2 не накладываются.

Таблица 7.1

Исходные данные к заданию

Первая цифра шифра	Коэффициенты функции цели	Вторая цифра шифра	Ограничения	Третья цифра шифра	d
1	2	3	4	5	6
0	$c_1 = d$ $c_2 = 8$	0	$3x_1 + 5x_2 \geq 2(2 + d)$ $2x_1 - x_2 \geq 2$ $x_1 - x_2 \leq 3$	0	2,0
1	$c_1 = -2$ $c_2 = 2d$	1	$-2x_1 + 1,5x_2 \geq 6$ $x_1 \leq -1$ $x_1 + x_2 \geq 2(2 - d)$ $3x_1 + 2x_2 \leq 12$	1	2,5
2	$c_1 = 1,5d$ $c_2 = -3$	2	$4x_1 + 3x_2 \leq -12$ $2dx_2 \leq 6$ $x_1 \geq -7$ $-2x_1 + x_2 \leq 4$	2	3,0
3	$c_1 = -4$ $c_2 = -2,5d$	3	$x_1 - 4x_2 \geq 0$ $-3x_1 + 2x_2 \leq 12$ $x_1 \leq d - 2$	3	3,5
4	$c_1 = 2d$ $c_2 = 5$	4	$x_1 \geq -1$ $2x_1 - 3x_2 \geq 6$ $x_1 + x_2 \geq -5$ $4x_1 + x_2 \leq 20 - d$	4	4,0

Окончание табл. 7.1

1	2	3	4	5	6
5	$c_1 = -6$ $c_2 = 2d$	5	$-x_1 - 2x_2 \leq 8$ $x_1 - 2x_2 \geq -6$ $dx_2 \leq 10$ $7x_1 - 4x_2 \leq 28$	5	3,5
6	$c_1 = d$ $c_2 = -7$	6	$-dx_1 + x_2 \leq 0$ $x_1 - 4x_2 \geq 0$ $x_1 - 2x_2 \leq 8$	6	3,0
7	$c_1 = -8$ $c_2 = -3d$	7	$2x_1 + x_2 \geq 6$ $x_1 - x_2 \leq 2$ $x_1 \geq 3 - d$	7	2,5
8	$c_1 = 6$ $c_2 = -d$	8	$2x_1 - 3x_2 \leq 6$ $x_1 + x_2 \geq 3$ $-x_1 + x_2 \leq 2$ $x_2 \leq d + 3$	8	2,0
9	$c_1 = -5$ $c_2 = 2d$	9	$-x_1 + 2x_2 \leq 10$ $x_1 + x_2 \geq d$ $2x_1 - x_2 \leq 4$	9	4,0

ЛАБОРАТОРНАЯ РАБОТА № 8 СИМПЛЕКС-МЕТОД РЕШЕНИЯ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Цель работы: изучить симплекс-метод, предназначенный для решения задач линейного программирования.

Состав работы:

1. Выводы по графическому решению ЗЛП. Идея и основные этапы симплекс-метода;
2. Жордановы исключения;
3. Приведение ЗЛП к каноническому виду;
4. Пример решения ЗЛП симплекс-методом;
5. Выполнение индивидуальных заданий.

8.1. Выводы по графическому решению ЗЛП. Идея и основные этапы симплекс-метода

Графическое решение задач с двумя переменными позволяет сделать два важных вывода:

- 1) ЗЛП может:
 - а) иметь единственное решение;
 - б) иметь бесконечное множество решений;
 - в) не иметь решения.
- 2) Если ЗЛП имеет решение, то оно достигается в одной или нескольких вершинах ОДР.

Распространяя эти выводы на ЗЛП с n переменными, можно предложить, казалось бы, следующий простой алгоритм: надо найти координаты всех вершин ОДР, вычислить во всех вершинах значения функции цели и, сравнив их между собой, выбрать ту из них, где функция цели наибольшая или наименьшая. Но дело в том, что в задаче с n переменными ОДР представляет собой сложный геометрический объект, ограниченный уже не отрезками прямых, а так называемыми гиперплоскостями. Число вершин в таком объекте и сложность определения их координат столь велики, что это не под силу сделать даже компьютеру.

Из стремления обойти эти трудности и возник универсальный метод решения ЗЛП с любым числом переменных, а именно,

симплекс-метод. Суть его заключается в следующем. Вначале, используя специальный алгоритм, находят любую вершину ОДР (ее координаты называются начальным планом). Применяя некоторые формальные признаки, эту вершину исследуют на оптимальность. Если оказывается, что оптимальной она не является, то опять-таки по строго определенным правилам переходят в соседнюю вершину, функция цели в которой больше или, по крайней мере, равна тому значению, которое она имела в начальной вершине. Новую вершину также проверяют на оптимальность и т. д. Процесс продолжается до тех пор, пока не будет достигнута оптимальная вершина либо не будет доказано, что поставленная задача решения не имеет. Опыт показывает, что происходит в итоге сравнительно небольшого числа шагов.

Симплекс-метод состоит из трех основных этапов:

- 1) приведение ЗЛП к каноническому виду и запись ее в начальную жорданову таблицу;
- 2) нахождение начального плана;
- 3) нахождение оптимального плана.

Каждый из этих этапов будет подробно рассмотрен ниже. Однако вначале познакомимся с аппаратом жордановых исключений, на котором базируются второй и третий этапы симплекс-метода.

8.2. Жордановы исключения

Пусть имеется система двух уравнений

$$\begin{aligned}y_1 &= 2x_1 - 5x_2 + 4x_3, \\y_2 &= 8x_1 + 2x_2 - 3x_3.\end{aligned}$$

Переменные x_1, x_2, x_3 , стоящие в правой части уравнений, назовем свободными или независимыми. Они могут принимать любые значения. Переменные y_1, y_2 , находящиеся в левой части, наоборот, будут зависимыми или несвободными. Их значения определяются выбором свободных переменных. Например, если принять $x_1 = x_2 = x_3 = 0$, то и y_1, y_2 будут равны нулю. А если $x_1 = 1, x_2 = 2, x_3 = 3$, то $y_1 = 4, y_2 = 3$.

Поставим задачу поменять местами какие-нибудь свободные и несвободные переменные, например, x_2 и y_2 . Такая замена и составляет суть жордановых исключений.

Разрешим второе уравнение относительно x_2

$$x_2 = -4x_1 + 0,5y_2 + 1,5x_3$$

и подставим это выражение в первое уравнение

$$y_1 = 2x_1 - 5(-4x_1 + 0,5y_2 + 1,5x_3) + 4x_3 = 22x_1 - 2,5y_2 - 3,5x_3.$$

Получили преобразованную систему

$$\begin{aligned} y_1 &= 22x_1 - 2,5y_2 - 3,5x_3, \\ x_2 &= -4x_1 + 0,5y_2 + 1,5x_3. \end{aligned}$$

в которой свободными переменными выступают уже x_1 , y_2 , x_3 , а несвободными – y_1 и x_2 .

Запишем исходную и преобразованную системы в жордановы таблицы:

	$-x_1$	$-x_2$	$-x_3$	
1	$y_1 =$	-2	5	-4
②	$y_2 =$	-8	-2	3
		1	②	3

	$-x_1$	$-y_2$	$-x_3$	
	$y_1 =$	-22	2,5	3,5
	$x_2 =$	4	-0,5	-1,5
		1	2	3

Обе таблицы имеют верх, в котором помещены свободные переменные, взятые со знаком минус. Такая форма их представления удобна для последующего использования в симплекс-методе. Обе таблицы имеют левый столбец, в котором располагаются зависимые переменные. Коэффициенты систем помещены в две строки и три столбца; их номера расположены слева и снизу от таблиц. Соответственно этим номерам будем считать, что коэффициентами исходной таблицы являются $a_{11} = -2$, $a_{12} = 5$, ... , а преобразованной – $a^*_{11} = -22$, $a^*_{12} = -2,5$,

Тот столбец ② из которого переносится независимая переменная x_2 , будем называть **разрешающим столбцом**. Ту строку ② из которого переносится независимая переменная y_2 , назовем **разрешающей строкой**. (Номера этой строки и столбца обведены кружком). На пересечении разрешающей строки и разрешающего столбца находится **разрешающий элемент** $a_{22} = -2$ (он выделен).

Чтобы проверить, что таблицы составлены правильно, можно от них перейти назад к уравнениям. Для этого коэффициенты любой строки надо умножить на стоящие над ними свободные переменные (с учетом их знака!), произведения сложить, а сумму приравнять соответствующей зависимой переменной. Например, из первой строки исходной таблицы получаем первое уравнение исходной системы

$$y_1 = -2 \cdot (-x_1) + 5 \cdot (-x_2) - 4 \cdot (-x_3) = 2x_1 - 5x_2 + 4x_3.$$

Сопоставляя исходную и преобразованную таблицы, можно непосредственно убедиться в справедливости следующих четырех правил, по которым происходит преобразование коэффициентов таблиц при жордановых исключениях

1) разрешающий элемент заменяется обратной величиной

$$a_{22} = -2; \quad a_{22}^* = 1/a_{22} = 1/(-2) = -0,5;$$

2) остальные элементы разрешающей строки делятся на разрешающий элемент

$$\begin{aligned} a_{21} &= -8, & a_{21}^* &= -8/(-2) = 4, \\ a_{23} &= 3, & a_{23}^* &= 3/(-2) = -1,5; \end{aligned}$$

3) остальные элементы разрешающего столбца делятся на разрешающий элемент и меняют знак на противоположный

$$a_{12} = 5, \quad a_{12}^* = -5/(-2) = 2,5;$$

4) все оставшиеся элементы преобразуются по правилу прямоугольника (рис. 8.1):

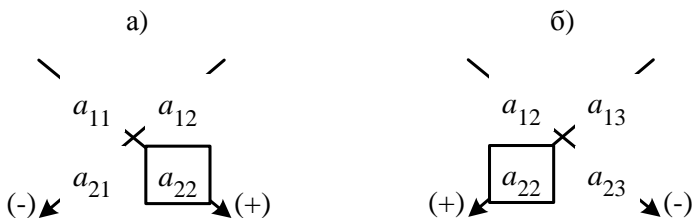


Рис. 8.1. Пояснение правила прямоугольников

а) элемент $a_{11} = -2$ входит в прямоугольник (рис. 8.1 а) в котором он образует положительную диагональ с разрешающим элементом a_{22} . Вторая диагональ $a_{21} - a_{12}$ считается отрицательной. Поэтому

$$a_{11}^* = \frac{a_{11}a_{22} - a_{12}a_{21}}{a_{22}} = \frac{-2 \cdot (-2) - 5 \cdot (-8)}{-2} = -22;$$

б) элемент $a_{13} = -4$ входит в другой прямоугольник (рис. 8.1 б), образуя положительную диагональ другого направления, но опять-таки с разрешающим элементом a_{22} . Поэтому

$$a_{13}^* = \frac{a_{13}a_{22} - a_{12}a_{23}}{a_{22}} = \frac{-4 \cdot (-2) - 5 \cdot 3}{-2} = 3,5.$$

В заключении отметим еще одно важное свойство изложенного варианта жордановых исключений. Независимая переменная, находясь на верху таблицы, имеет знак минус, а при переходе в левый столбец теряет его. Зависимая переменная, переходя из левого столбца наверх, знак минус приобретает.

8.3. Приведение ЗЛП к каноническому виду

Для того, чтобы придать симплекс-методу универсальный характер, необходимо привести решаемую задачу к каноническому виду.

ЗЛП считается записанной в *каноническом виде*, если она удовлетворяет четырем требованиям:

- 1) Функция цели максимизируется.
- 2) Ограничения – строгие равенства.
- 3) Свободные члены ограничений неотрицательны.
- 4) Условия неотрицательности наложены на все переменные.

8.4. Пример решения ЗЛП симплекс-методом

Рассмотрим задачу из предыдущей лабораторной работы:

$$Z = x_1 - 2x_2 \rightarrow \begin{matrix} \min \\ \max \end{matrix},$$

$$3x_1 + 4x_2 \leq 12,$$

$$2x_1 - x_2 \geq -2,$$

$$x_1 - 2x_2 \leq 6,$$

$$x_1 \geq 0.$$

При такой постановке эта задача состоит из двух отдельных задач: минимизации Z и максимизации Z .

Мы будем рассматривать первую из них – минимизации Z :

$$Z = x_1 - 2x_2 \rightarrow \min,$$

$$3x_1 + 4x_2 \leq 12,$$

$$2x_1 - x_2 \geq -2,$$

$$x_1 - 2x_2 \leq 6,$$

$$x_1 \geq 0.$$

Сравнив условие задачи и требования при записи канонической формы, видно, что в ней нарушаются все четыре правила. Устраним их.

1) Чтобы от задачи минимизации перейти к задаче на максимум, надо функцию цели умножить на -1 . Тогда первая часть исходной задачи в новой постановке будет выглядеть так

$$(-Z) = -x_1 + 2x_2 \rightarrow \max.$$

После решения новой задачи и определения $\max(-Z)$ минимальное значение Z определяется как $\min(Z) = -\max(-Z)$.

2) Если ограничение содержит знак \leq , то к его левой части нужно добавить некоторую новую неотрицательную переменную. Тогда вместо заданного ограничения мы получаем две записи, включающие собственно ограничение со знаком $-$ и условие неотрицательности дополнительной переменной. Соответственно,

если ограничение имеет знак \geq , то от левой части отнимается новая неотрицательная переменная.

Преобразуем ограничения в исходной задаче:

$$3x_1 + 4x_2 \leq 12 \Rightarrow 3x_1 + 4x_2 + x_3 = 12, \text{ где } x_3 \geq 0,$$

$$2x_1 - x_2 \geq -2 \Rightarrow 2x_1 - x_2 - x_4 = -2, \text{ где } x_4 \geq 0,$$

$$x_1 - 2x_2 \leq 6 \Rightarrow x_1 - 2x_2 + x_5 \leq 6, \text{ где } x_5 \geq 0.$$

3) Если ограничение имеет отрицательный свободный член, то все оно умножается на -1 . В данном случае изменит свой вид второе ограничение:

$$-2x_1 + x_2 + x_4 = 2, x_4 \geq 0$$

4) Если на какую-нибудь переменную не наложено условие неотрицательности, то она заменяется разностью двух неотрицательных величин.

В исходной задаче условие неотрицательности задано только для переменной x_1 . Для переменной x_2 такое условие не указано.

Тогда

$$x_2 = x_2^{(1)} - x_2^{(2)}, \text{ где } x_2^{(1)} \geq 0 \text{ и } x_2^{(2)} \geq 0.$$

Собирая и редактируя снова всю задачу, получаем окончательно

$$(-Z) = -x_1 + 2x_2^{(1)} - 2x_2^{(2)} \rightarrow \max,$$

$$3x_1 + 4x_2^{(1)} - 4x_2^{(2)} + x_3 = 12,$$

$$-2x_1 + x_2^{(1)} - x_2^{(2)} + x_4 = 2,$$

$$x_1 - 2x_2^{(1)} + 2x_2^{(2)} + x_5 = 6,$$

$$x_1 \geq 0, x_2^{(1)} \geq 0, x_2^{(2)} \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0.$$

Видно, что записав ЗЛП в каноническом виде, изменилась размерность задачи: число неизвестных возросло с двух до шести.

8.4.1. Запись начальной жордановой таблицы

Чтобы записать начальную жорданову таблицу, преобразуем исходную задачу еще раз, представив функцию цели в виде

$$(-Z) = 1 \cdot (-x_1) - 2 \cdot (-x_2^{(1)}) + 2(-x_2^{(2)}) + 0 \cdot (-x_3) + 0 \cdot (-x_4) + 0 \cdot (-x_5),$$

а ограничения в виде так называемых нуль-строк

$$0 = 12 + 3 \cdot (-x_1) + 4 \cdot (-x_2^{(1)}) + (-4) \cdot (-x_2^{(2)}) + 1 \cdot (-x_3),$$

$$0 = 2 - 2 \cdot (-x_1) + 1 \cdot (-x_2^{(1)}) - 1 \cdot (-x_2^{(2)}) + 1 \cdot (-x_4),$$

$$0 = 6 + 1 \cdot (-x_1) - 2 \cdot (-x_2^{(1)}) + 2 \cdot (-x_2^{(2)}) + 1 \cdot (-x_5).$$

Нетрудно видеть, что получены они переносом всех слагаемых в правую часть ограничений.

Начальная жорданова таблица (табл. 8.1) состоит из верха, где расположены все шесть переменных, трех нуль-строк, соответствующих трем ограничениям, и Z-строки. В левом столбце помещены нули и наименование функции цели. Во втором столбце – свободные члены ограничений и функции цели (последний в начальной таблице обычно равен нулю). Условия неотрицательности переменных в таблицу не заносятся.

Таблица 8.1

Начальная жорданова таблица

		$-x_1$	$-x_2^{(1)}$	$-x_2^{(2)}$	$-x_3$	$-x_4$	$-x_5$
$0=$	12	3	4	-4	1	0	0
$0=$	2	-2	1	-1	0	1	0
$0=$	6	1	-2	2	0	0	1
$(-Z)=$	0	1	-2	2	0	0	0

8.4.2. Нахождение начального плана

Начальный план – это координаты любой вершины ОДР. Чтобы его найти, надо все нули в левом столбце заместить переменными с верха таблицы. Делается это с помощью жордановых исключений.

Чтобы некоторая переменная могла быть перенесена, в ее столбце (исключая элемент Z -строки) должен быть положительный коэффициент; он принимается за разрешающий элемент. Если в столбце два и более положительных коэффициентов, то за разрешающий элемент берется тот, который соответствует минимальному отношению свободного члена к коэффициенту столбца. И еще: место в левом столбце, куда собираются перенести неизвестную, не должно быть занято ранее перенесенной неизвестной.

При переносе переменной с верха таблицы на ее место должен стать нуль. Это означает умножение столбца на нуль. Поэтому в новой таблице этот столбец вычеркивается. Элементы остальных столбцов и Z -строки преобразуются по второму и четвертому правилам жордановых исключений (см. п. 8.2).

В нашей задаче (табл. 8.1) переменная x_1 может быть перенесена в первую или третью строки (надо уточнить, в какую именно), $x_2^{(1)}$ – в первую либо вторую, $x_2^{(2)}$ – только в третью, x_3 – в первую, x_4 – во вторую, x_5 – в третью.

Начнем с x_1 . В столбце этой переменной два положительных коэффициента: 3 и 1. Один из них должен быть взят за разрешающий элемент. Но какой именно? Разделим свободный член первой строки 12 на 3, получим 4. А частное от деления свободного члена третьей строки 6 на 1 равно 6, т. е. больше. Поэтому за разрешающий элемент берем 3 (в табл. 8.1 он выделен). Это значит, что x_1 переносится в первую строку.

Первая жорданова таблица (табл. 8.2) имеет на один столбец меньше. Коэффициенты первой (разрешающей) строки получены по второму правилу жордановых исключений. Коэффициенты второй, третьей и Z -строки преобразованы, как прочие элементы (правило 4). При вычислении коэффициентов первой жордановой таблицы использованы (для наглядности) простые дроби.

Таблица 8.2

Первая жорданова таблица

		$-x_2^{(1)}$	$-x_2^{(2)}$	$-x_3$	$-x_4$	$-x_5$
$x_1 =$	4	4/3	-4/3	1/3	0	0
$0 =$	10	11/3	-11/3	2/3	1	0
$0 =$	2	-10/3	10/3	-1/3	0	1
$(-Z) =$	-4	-10/3	10/3	-1/3	0	0

Далее попробуем перенести $x_2^{(2)}$. В соответствующем столбце один положительный коэффициент (10/3), и место в третьей строке не занято. Следовательно, такой перенос возможен. Используя разрешающий элемент 10/3 (в табл. 8.2 он выделен), строим вторую жорданову таблицу (табл. 8.3).

Таблица 8.3

Вторая жорданова таблица.

		$-x_2^{(1)}$	$-x_3$	$-x_4$	$-x_5$
$x_1 =$	4,8	0	0,2	0	0,4
$0 =$	12,2	0	0,3	1	1,1
$x_2^{(2)} =$	0,6	-1	-0,1	0	0,3
$(-Z) =$	-6	0	0	0	-1

Остается заместить нуль во второй строке. Проще всего это сделать, перенеся сюда x_4 : в соответствующем столбце – один положительный коэффициент и именно во второй строке. Принимаем его за разрешающий и строим третью жорданову таблицу (табл. 8.4).

Таблица 8.4

Третья жорданова таблица

		$-x_2^{(1)}$	$-x_3$	$-x_5$
$x_1 =$	4,8	0	0,2	0,4
$x_4 =$	12,2	0	0,3	1,1
$x_2^{(2)} =$	0,6	-1	-0,1	0,3
$(-Z) =$	-6	0	0	-1

Итак, все нули в левом столбце замещены. Мы пришли в одну из вершин ОДР, называемую начальной. Чтобы найти ее координаты (начальный план), надо перенесенные в левый столбец переменные приравнять соответствующим свободным членам ограничений ($x_1 = 4,8$; $x_4 = 12,2$; $x_2^{(2)} = 0,6$), а оставшиеся наверху таблицы свободные переменные приравнять нулю ($x_2^{(1)} = x_3 = x_5 = 0$). Функция цели в этой вершине ($-Z = -6$).

Следует отметить, что если бы мы изменили принятый выше порядок переноса неизвестных, то могли бы прийти в другую вершину ОДР.

8.4.3. Нахождение оптимального плана

Достигнутую вершину или план $x_1 = 4,8$; $x_2^{(1)} = 0$; $x_2^{(2)} = 0,6$; $x_3 = 0$; $x_4 = 12,2$; $x_5 = 0$, при котором ($-Z = -6$), надо исследовать на оптимальность. Делается это очень просто. Надо просмотреть коэффициенты Z – строки (в табл. 8.4 это 0; 0; -1). Если среди них имеется хотя бы один отрицательный (а у нас он есть и равен -1), то достигнутая вершина и соответствующий ей план не является оптимальным.

Нужно перейти к другой вершине. В качестве разрешающего столбца берется тот столбец, в котором находится отрицательный коэффициент функции цели. У нас это последний столбец.

В нем три положительных коэффициента. (Если бы в этом столбце не оказалось ни одного положительного коэффициента, то это свидетельствовало бы о том, что данная задача решения не имеет). Из трех положительных коэффициентов один надо выбрать в качестве разрешающего элемента. Для этого находим три частных: $4,8/0,4 = 12$; $12,2/1,1 = 11,0909$; $0,6/0,3 = 2$. Наименьшее из них – последнее. Отсюда разрешающая строка – третья. На ее пересечении с разрешающим столбцом стоит разрешающий элемент 0,3, выделенный в табл. 8.4.

Переходим к четвертой жордановой таблице (табл. 8.5), в которой неизвестная x_5 перешла в левый столбец, а на ее место стала $x_2^{(2)}$. Коэффициенты новой таблицы находятся по четырем правилам жордановых исключений.

Таблица 8.5

Четвертая жорданова таблица

		$-x_2^{(1)}$	$-x_3$	$-x_2^{(2)}$
$x_1 =$	4	4/3	1/3	-4/3
$x_4 =$	10	11/3	2/3	-11/3
$x_5 =$	2	-10/3	-1/3	10/3
$(-Z) =$	-4	-10/3	-1/3	10/3

Достигли новой вершины с координатами $x_1 = 4$; $x_2^{(1)} = 0$; $x_2^{(2)} = 0$; $x_3 = 0$; $x_4 = 10$; $x_5 = 2$. Функция цели $(-Z)$ выросла с -6 до -4. Этот план также не является оптимальным, так как среди коэффициентов Z – строки имеются отрицательные (-10/3 и -1/3).

Надо перейти к следующей вершине ОДР. Поскольку в Z – строке два отрицательных коэффициента, то за разрешающий столбец берем тот, в котором стоит отрицательный коэффициент с большим модулем (-10/3). В разрешающем столбце имеются два положительных коэффициента. Находим частные $4 \cdot 3/4 = 3$ и $10 \cdot 3/11 = 2,72727$. Вторая величина меньше. Поэтому разрешающая строка – вторая, разрешающий элемент равен 11/3.

Переходим к пятой таблице (табл. 8.6), меняя местами x_4 и $x_2^{(1)}$.

Таблица 8.6

Пятая жорданова таблица

		$-x_4$	$-x_3$	$-x_2^{(2)}$
$x_1 =$	4/11	-4/3	1/11	0
$x_2^{(1)} =$	30/11	3/11	2/11	-1
$x_5 =$	122/11	10/11	3/11	0
$(-Z) =$	56/11	10/11	3/11	0

Достигли вершины с координатами $x_1 = 4/11$; $x_2^{(1)} = 30/11$; $x_2^{(2)} = 0$; $x_3 = 0$; $x_4 = 0$; $x_5 = 122/11$. Функция цели выросла до $(-Z) = 56/11$. Т.к. среди коэффициентов Z – строки нет отрицательных, полученный план является оптимальным. Переходя к исходным неизвестным, получаем окончательный результат

$$x_1 = 4/11 = 0,363636; x_2 = x_2^{(1)} - x_2^{(2)} = 30/11 = 2,72727,$$

$$\min Z = -\max (-Z) = -56/11 = -5,09091.$$

8.5. Выполнение индивидуальных заданий

Используя исходные данные к лабораторной работе № 7, решить свой вариант задания, применяя симплекс-метод.

Указания к работе.

Функция цели в задаче исследуется на максимум. Условия неотрицательности на переменные x_1 и x_2 не накладываются.

Литература

1. Бахвалов, Н.С. Численные методы (анализ, алгебра, обыкновенные дифференциальные уравнения) / Н.С. Бахвалов. – М.: Наука, 1975. – 630 с.
2. Турчак, Л.И. Основы численных методов / Л.И. Турчак. – М.: Наука, 1987. – 320 с.
3. Крылов, В.И., Вычислительные методы: в 2-х томах / В.И. Крылов, В.В. Бобков, П.И. Монастырский. – М.: Наука. – Т.1, 1976. – 333 с. – Т.2, 1977. – 399 с.
4. Караманский, Т. Д. Численные методы строительной механики / Т. Д. Караманский. – М.: Стройиздат, 1981. – 428 с.
5. Амосов, А.А. Вычислительные методы для инженеров: учебное пособие / А. А. Амосов, Ю А Дубинский, Н. В. Копченко – М.: Высш. школа, 1994. – 544 с.: ил.
6. Костевич, Л.С. Математическое программирование: Информ. технологии оптимальных решений: учеб. пособие / Л.С. Костевич. Минск: Новое знание. – 2003. – 424 с.: ил.
7. Кузнецов, А.В., Математическое программирование / А.В. Кузнецов, Н.И. Холод. – Минск: Высшая школа, 1984. – 221 с.
8. Мудров, А.Е. Численные методы для ПЭВМ на языках Бейсик, Фортран и Паскаль / А.Е. Мудров. – Томск: МП «РАСКО». – 1991. – 272 с.: ил.
9. Дьяконов, В.П. Mathematica 5/6/7. Полное руководство / В.П. Дьяконов. – М.: ДМК, Пресс, 2009. – 624 с.
10. Половко, А.М. Mathematica для студента / А.М. Половко. – СПб.: БХВ-Петербург. – 2007. – 368 с.: ил.

$$\mathbf{O} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{bmatrix}.$$

Подобно системам уравнений, квадратные матрицы могут быть *симметричными* (элементы расположены симметрично относительно главной диагонали), *треугольными* (элементы, равные нулю, расположены ниже (выше) главной диагонали), *ленточными* (ненулевые элементы образуют «ленту», параллельную диагонали) и т. д.

Определителем n -го порядка называется число Δ , которое символически записывается с помощью квадратной таблицы

$$\Delta = \det |\mathbf{A}| = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{vmatrix}$$

и вычисляется следующим образом

$$n = 1, \Delta = |a_{11}| = a_{11},$$

$$n = 2, \Delta = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21},$$

$$n = 3; \Delta = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}.$$

Суммой (разностью) двух матриц \mathbf{A} и \mathbf{B} одинаковых размеров называется третья матрица \mathbf{C} тех же размеров, элементы которой определяются по формуле

$$c_{ik} = a_{ik} \pm b_{ik}.$$

Транспонирование матрицы – это замена строк столбцами и наоборот. В публикациях операция транспонирования используется для более удобной записи векторов. Например, вместо того, чтобы изображать вектор в виде

$$\vec{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

удобнее представить его как $\vec{x} = [1 \ 2 \ 3]^T$.

При **умножении** (или **делении**) произвольной **матрицы на число**, отличное от нуля, получается матрица тех же размеров, все элементы которой получены умножением (или делением) элементов исходной матрицы на это число.

Две матрицы **A** и **B** можно перемножить между собой и получить третью матрицу **C** в том и только в том случае, когда число столбцов матрицы **A** равно числу строк матрицы **B**. Результирующая матрица имеет число строк, как у матрицы **A**, а число столбцов – как у **B**: $\mathbf{A} \cdot \mathbf{B} = \mathbf{C}$.

$$m \times n \quad n \times p \quad m \times p$$

Элементы результирующей матрицы **C** определяются по формуле $c_{ik} = \sum_{j=1}^n a_{ij} \cdot b_{jk}$, т. е. элементы *i*-й строки матрицы **A** перемножаются

на соответствующие элементы *k*-го столбца матрицы **B** и все результаты суммируются.

Матричное произведение в общем случае не обладает перестановочным свойством. Так, если матрицы **A** и **B** из только что рассмотренного примера перемножить в обратном порядке **B**·**A**, то результирующая матрица **D** будет совершенно иной, чем **C**. Очень часто произведение двух матриц вообще невозможно либо возможно только в одном порядке. Если произведение двух матриц возможно как в прямом, так и в обратном порядке и дает один и тот же результат, то такие матрицы называются **перестановочными**. Примером перестановочной матрицы является единичная матрица.

Операция перемножения матриц позволяет ввести матричную запись системы линейных уравнений (СЛАУ)

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & & & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}$$

сокращенно в виде $\mathbf{A}\vec{x} = \vec{b}$ или $\mathbf{A}\mathbf{X} = \mathbf{B}$, где \mathbf{A} – матрица коэффициентов; $\vec{x}(\mathbf{X})$ – вектор (матрица) неизвестных; $\vec{b}(\mathbf{B})$ – вектор (матрица) свободных членов.

Матрицей, **обратной** матрице \mathbf{A} , называется такая матрица \mathbf{A}^{-1} , которая удовлетворяет соотношениям

$$\underset{n \times n}{\mathbf{A}} \cdot \underset{n \times n}{\mathbf{A}^{-1}} = \underset{n \times n}{\mathbf{A}^{-1}} \cdot \underset{n \times n}{\mathbf{A}} = \underset{n \times n}{\mathbf{E}} .$$

Для того, чтобы обратить квадратную матрицу, она должна быть **невырожденной** (определитель не равнялся нулю).

Невырожденная матрица \mathbf{A} является **плохо обусловленной**, если ее определитель относительно мал, т. е. $\det \mathbf{A} \approx 0$. Использование таких матриц в приближенных вычислениях приводит, как правило, к получению неточных и неустойчивых результатов.

Учебное издание

**ЧИСЛЕННЫЕ МЕТОДЫ
РЕШЕНИЯ ЗАДАЧ СТРОИТЕЛЬСТВА**

Лабораторный практикум
для студентов специальности 1-70 02 01
«Промышленное и гражданское строительство»

Составители:

СТРЕЛЮХИН Александр Владиславович
БОГОМОЛОВА Гелена Станиславовна
СОРОКИНА Елена Леонтьевна

Редактор *О. В. Ткачук*
Компьютерная верстка *Ю. С. Кругловой, А. Е. Дарвина*

Подписано в печать 31.05.2016. Формат 60×84 ¹/₁₆. Бумага офсетная. Ризография.
Усл. печ. л. 6,74. Уч.-изд. л. 5,27. Тираж 100. Заказ 468.

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