

КАЗАНСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

К.Н. Стехина, Д.Н. Тумаков

РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ
В ПАКЕТЕ MATHEMATICA

Часть 1. Уравнения первого порядка и их приложения

Учебное пособие

КАЗАНЬ
2014

УДК 681.3, 517.9

ББК 22.1

*Печатается по решению Редакционно-издательского совета
ФГАОУВПО «Казанский (Приволжский) федеральный университет»*

*методической комиссии ИВМиИТ
Протокол № 3 от 12 ноября 2014 г.*

*заседания кафедры прикладной математики
Протокол № 3 от 13 ноября 2014 г.*

*Научный редактор
доктор физ.-мат. наук, проф. Н. Б. Плещинский*

*Рецензенты
кандидат физ.-мат. наук, доцент И.Е. Плещинская
кандидат физ.-мат. наук, доцент Е.В. Рунг*

Стехина К.Н.

Решение дифференциальных уравнений в пакете *Mathematica*. Часть 1.
Уравнения первого порядка и их приложения: учебное пособие / К.Н. Стхина,
Д.Н. Тумаков. – Казань, 2014. – 116 с.

В пособии рассматривается применение системы *Mathematica* к решению обыкновенных дифференциальных уравнений. Дается краткая характеристика пакета, а также описание основных встроенных функций и особенностей их использования. Приведены примеры решения основных типов обыкновенных дифференциальных уравнений, с помощью рассматриваемого пакета с иллюстрацией основных моментов и особенностей. Показано использование графических возможностей пакета *Mathematica* для визуализации получаемых решений.

Пособие предназначено для студентов, магистрантов и аспирантов ВУЗов, специализирующихся по дифференциальным уравнениям, математическому моделированию и их приложениям.

© Казанский университет, 2014
© Стхина К.Н., Тумаков Д.Н., 2014

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
ГЛАВА 1. ОСНОВЫ ПАКЕТА MATHEMATICA	6
§1. ВВЕДЕНИЕ В ПАКЕТ MATHEMATICA	6
Запуск программы, работа, справка, выход.....	7
Основные операции и функции.....	11
Векторы, матрицы, списки, таблицы.....	12
Выражения, функции.....	13
§2. РЕШЕНИЕ УРАВНЕНИЙ	21
§3. ДИФФЕРЕНЦИРОВАНИЕ И РЕШЕНИЕ ПРОСТЕЙШИХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ.	27
Дифференцирование.....	27
Решение простейших дифференциальных уравнений	29
§4. ИНТЕГРИРОВАНИЕ ФУНКЦИЙ И ВЫЧИСЛЕНИЕ ИНТЕГРАЛОВ.....	33
§5. РАЗЛОЖЕНИЕ В РЯД И ВЫЧИСЛЕНИЕ ПРЕДЕЛОВ	37
§6. ПОСТРОЕНИЕ ГРАФИКОВ ФУНКЦИЙ	40
Графики функций одного аргумента.....	40
Визуализация функций двух аргументов.....	50
§7. НЕКОТОРЫЕ ЭЛЕМЕНТЫ ПРОГРАММИРОВАНИЯ	57
ГЛАВА 2. РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ.....	59
§1. УРАВНЕНИЯ С РАЗДЕЛЯЮЩИМИСЯ ПЕРЕМЕННЫМИ	60
§2. ОДНОРОДНЫЕ УРАВНЕНИЯ	66
§3. УРАВНЕНИЯ В ПОЛНЫХ ДИФФЕРЕНЦИАЛАХ	71
§4. ЛИНЕЙНЫЕ УРАВНЕНИЯ 1-ГО ПОРЯДКА.....	79
§5. НЕКОТОРЫЕ СПЕЦИАЛЬНЫЕ ДИФФЕРЕНЦИАЛЬНЫЕ УРАВНЕНИЯ ПЕРВОГО ПОРЯДКА.....	88
Уравнение Бернулли	88
Уравнение Клеро	91
Уравнение Лагранжа	97
Уравнение Риккати	99
ГЛАВА 3. ПРИЛОЖЕНИЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ 1-ГО ПОРЯДКА.....	103
§1. ОРТОГОНАЛЬНЫЕ ТРАЕКТОРИИ.....	103

§2. ПОЛЕ НАПРАВЛЕНИЙ	109
§3. ВОЗРАСТАЮЩИЕ И УБЫВАЮЩИЕ ПОПУЛЯЦИИ.....	111
§4. ЗАКОН НЬЮТОНА-РИХМАНА (О ТЕПЛООБМЕНЕ)	114
ЛИТЕРАТУРА	117

ВВЕДЕНИЕ

Система компьютерной алгебры *Mathematica*, разработанная американской компанией Wolfram Research Inc., является одним из наиболее распространенных программных средств, которое позволяет весьма эффективно выполнять как численные, так и символьные вычисления, имеет развитую двумерную и трехмерную графику, а также встроенный язык программирования высокого уровня.

Прикладной пакет *Mathematica* позволяет находить решение дифференциального уравнения или системы уравнений, как в символьном, так и в численном виде. Кроме того, есть возможность визуализации полученных результатов. Естественным образом любая поставленная математическая задача предварительно исследуется классическими методами теории познания, такими как анализ и синтез, индукция и дедукция, а затем происходит ее формализация и алгоритмическая реализация. Наличие языка программирования в прикладном пакете *Mathematica* позволяет составлять программы для широкого класса задач, в которых исходные данные можно свободно варьировать и проводить крупномасштабные эксперименты, подтверждая или опровергая выдвинутые гипотезы. Это также дает возможность исследовать реальные физические процессы с целью их прогнозирования.

Основная цель предлагаемого пособия – познакомить пользователей с пакетом *Mathematica* и его возможностями и научить его применению для решения основных типов обыкновенных дифференциальных уравнений и их систем. Следует отметить, что во многих случаях такие решения могут быть найдены с помощью соответствующих встроенных в пакет *Mathematica* функций. Поэтому в пособии детально рассматриваются методы интегрирования дифференциальных уравнений с использованием пакета *Mathematica* для символьных вычислений и преобразований уравнений к виду, удобному для интегрирования. Многочисленные примеры показывают, что при объединении теории дифференциальных уравнений с возможностями пакета *Mathematica* удается проинтегрировать даже такие уравнения, которые не решаются непосредственно с помощью встроенных функций.

ГЛАВА 1. ОСНОВЫ ПАКЕТА MATHEMATICA

§1. ВВЕДЕНИЕ В ПАКЕТ MATHEMATICA

Пакет *Mathematica* является универсальной технической компьютерной системой, обладающей возможностями компьютерной математики, имеющей свой язык программирования, инструменты публикации, разнообразные графические возможности, а также высокий уровень интеграции между всеми этими компонентами.

Важным аспектом архитектуры среды *Mathematica* является независимость от платформы, то есть пользователь может запускать ее на необычайно широком круге операционных систем. *Mathematica* доступна под Windows95/98/2000/NT/XP/Vista/7/8, MacOS, Linux, SunOs/Solaris, HP-UX, AIX, DigitalUnix и др.

Mathematica представляет собой программное средство для выполнения математических расчетов на компьютере. Пакет *Mathematica* состоит из двух частей: Ядро (*Kernel*) и Оболочка (*FrontEnd*). Ядро представляет собой программное обеспечение, непосредственно выполняющее расчеты, которое работает одинаково на всех типах компьютеров. Оболочка обеспечивает интерфейс между ядром и пользователем. В большинстве компьютерных систем используется интерфейс типа Notebook, который позволяет создавать документы, содержащие текст, графики, звук, активные формулы и команды, обрабатываемые ядром. Поскольку формат рабочих документов *Mathematica* — блокнотов, имеющих расширение .nb (от англ. "notebook"), — также независим от платформы, то пользователи любой из перечисленных систем могут обмениваться блокнотами *Mathematica*, просто копируя их или пересылая по электронной почте. Бесплатное приложение *MathReader* разрешает просмотр и печать .nb-документа, запуск анимации и т.д. Это делает .nb-стандарт идеальным информационным каналом, общим средством общения среди профессионалов всего мира.

Mathematica позволяет производить манипулирование алгебраическими формулами, то есть разлагать на множители, раскрывать скобки и производить упрощение полиномов и рациональных выражений. Она также позволяет находить алгебраические решения полиномиальных уравнений и систем уравнений, вычислять интегралы и производные, решать дифференциальные уравнения, представлять функции в виде разложения в ряд, а также вычислять пределы в символьной форме.

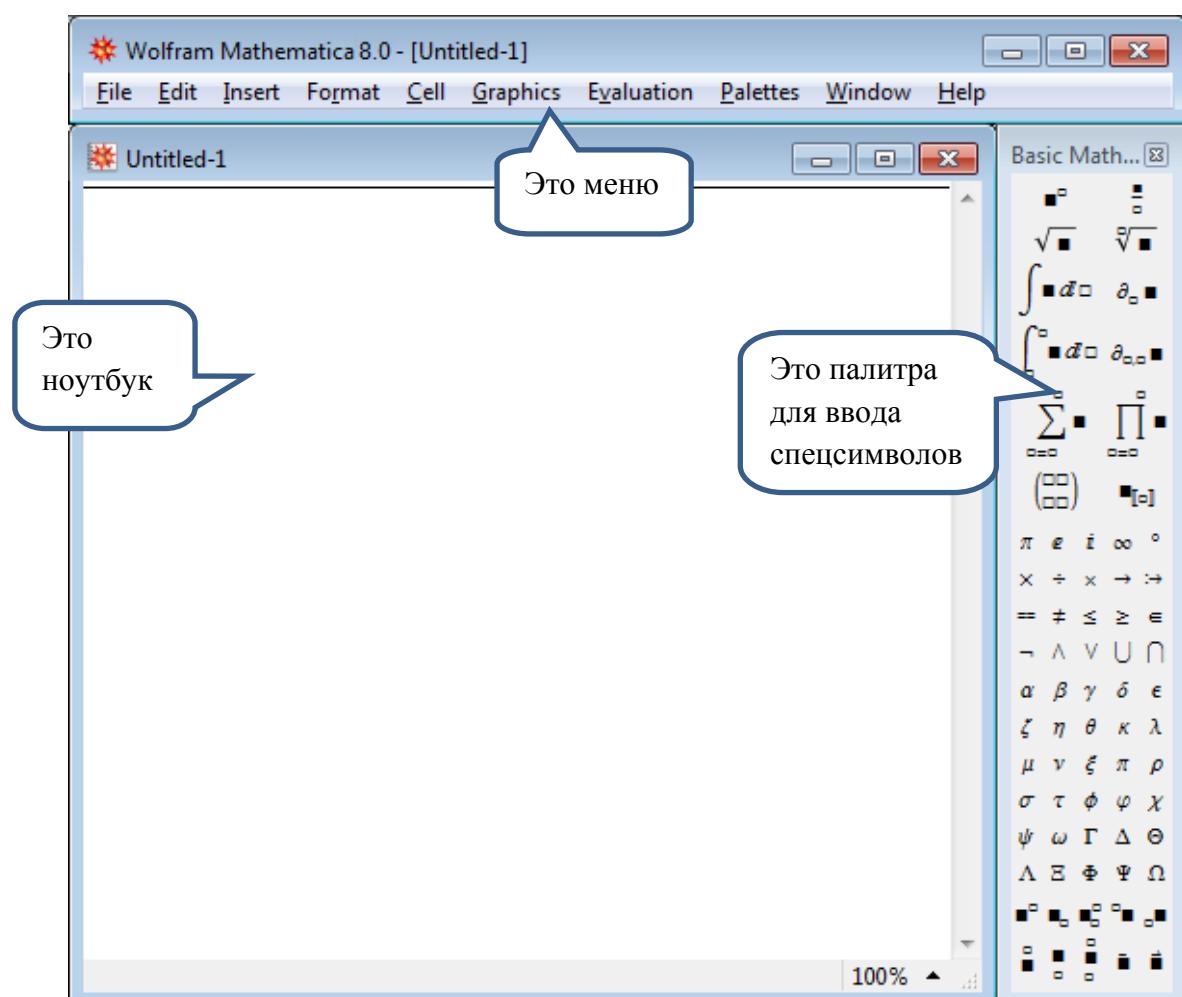
Mathematica также может производить расчеты с использованием большого числа специальных функций. Кроме того, *Mathematica* может производить численные вычисления с любой точностью. С помощью пакета *Mathematica* можно численно вычислять интегралы, решать алгебраические и дифференциальные уравнения и системы уравнений, обрабатывать численные данные, производя их статистический анализ, производить фурье-анализ,

интерполяцию и аппроксимацию данных с помощью метода наименьших квадратов. *Mathematica* может работать не только с числами, но и с матрицами, обеспечивая выполнение всех операций линейной алгебры.

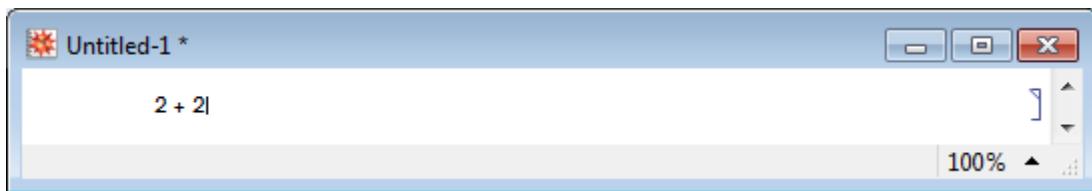
В пакете *Mathematica* можно строить двумерные и трехмерные графики функций, заданных явно или в параметрической форме, а также контурные графики и графики плотности. В пакете *Mathematica* существует много опций, позволяющих контролировать различные аспекты графиков, например, цвет и тип линий, тени, отображение осей координат и так далее.

Запуск программы, работа, справка, выход

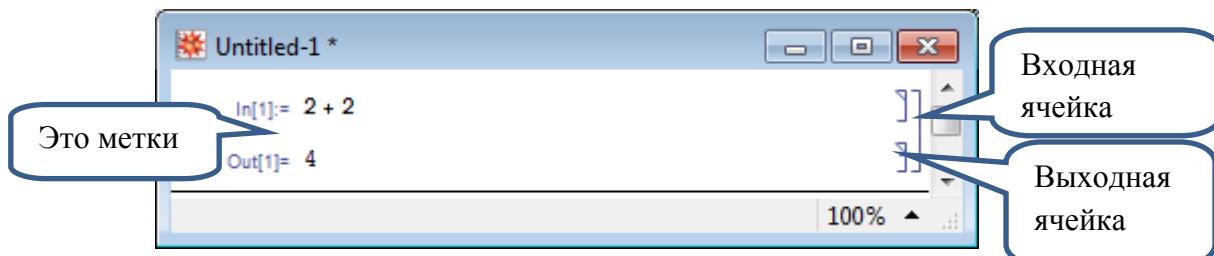
Для запуска программы *Mathematica* достаточно с помощью левой кнопки мыши щелкнуть иконку *Mathematica* в меню «Программы» или ярлык программы в месте его расположения. После нескольких секунд загрузки появляется рабочее окно и создается бланк нового документа



Напишем $2+2$. Введенные данные автоматически объединяются во входную ячейку, которая обозначается]-скобкой в правой части документа. Ячейка содержит, по крайней мере, одну строку текста и является наименьшей частью данных, которые обрабатываются ядром



Для обработки поместим курсор в любой части ячейки и нажмем **[Shift]+[Enter]** (нажатие одиночного **[Enter]** на основной клавиатуре приводит к созданию новой строки в той же ячейке). Если введенные данные являются логически завершенными и не содержат синтаксических ошибок, *Mathematica* обрабатывает их и выдает результат. В противном случае она перепечатает введенные данные и укажет тип ошибки. Ячейка в процессе вычисления принимает желтый цвет, а в заглавии окна появляется *Running*



Первое вычисление выполняется всегда дольше остальных, так как подгружается интеллектуальное ядро *Mathematica* (*Mathematica Kernel*).

Если вычисления делятся достаточно долго, и вы их хотите прервать, то воспользуйтесь комбинацией клавиш **[Alt]+[.]** или осуществите прерывание из меню *Evaluation*→.

Если же ничего не выходит, то попробуйте завершить работу *Ядра* системы *Mathematica* (*Evaluation*→*QuitKernel*→*Local*).

При обработке данных *Mathematica* снабжает их метками: *In[n]:=* -- команда или выражение, вводимые пользователем, *Out[n]=*--результат, выводимый программой *Mathematica*, где $n=1,2,3,\dots$. Номер n , которым помечается очередной диалог, может быть использован для ссылки на любой предыдущий результат. Так, знак $\%n$ соответствует n -му выведенному программой *Mathematica* результату.

В одной строке можно вводить несколько выражений, разделяя их точкой с запятой «;». Если в конце выражения стоит точка с запятой, то *Mathematica* обрабатывает его, но не выводит полученный результат.

Весь создаваемый документ имеет секционную структуру, то есть, состоит из набора секций. Каждая секция выделяется с помощью ограничивающей ее справа]-скобки и имеет свой тип, который определяет ее роль в данном документе. Например, секция, содержащая команды и формулы, предназначенные для обработки ядром, имеет тип *Input*. Соответствующий результат, полученный при обработке введенных данных, имеет тип *Output*. С другой стороны, текст, предназначенный для чтения, имеет тип *Text* и не обрабатывается ядром. По умолчанию каждая вновь создаваемая секция имеет тип *Input*. Чтобы изменить или определить тип какой-либо секции, необходимо сначала выделить ее, подведя указатель мыши к ограничивающей ее]-скобке и щелкнув один раз левой кнопкой мыши. Затем нужно с помощью мыши

щелкнуть в меню иконку *Format* и выбрать в разделе *Style* нужный тип. Две и более секции могут объединяться в группу. В этом случае они ограничиваются справа еще одной *]*-скобкой. Последовательность обработки секций в документе определяется пользователем.

Любому результату или выражению можно присвоить имя, которое затем будет использоваться для ссылок на этот результат. При этом следует учитывать, что *Mathematica* различает заглавные и строчные буквы.

Значение, присвоенное переменной, сохраняется в течение всего сеанса работы с программой *Mathematica*. Для удаления значения, присвоенного переменной, используется команда *Clear[]* или «=.=», причем символы «=.=» набираются один за другим без пробела

```
In[1]:= a = 5
Out[1]= 5

In[6]:= a = .
Out[6]= .

In[7]:= a
Out[7]= a

или

In[8]:= a = 10
Out[8]= 10

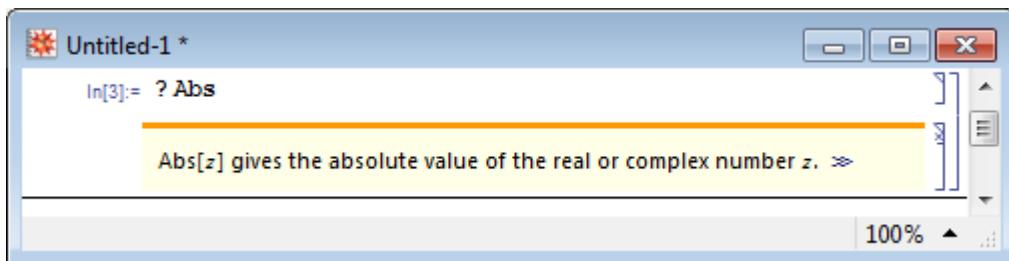
In[9]:= Clear[a]
Out[9]= a

In[10]:= a
Out[10]= a
```

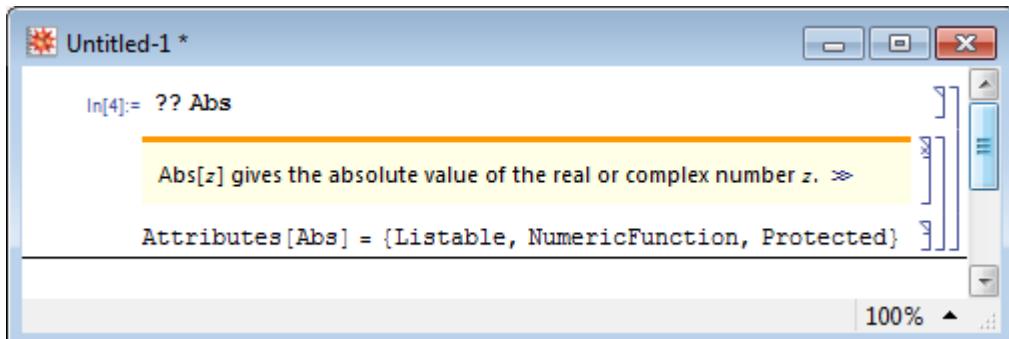
Приступая к решению новой задачи, полезно также удалить все введенные ранее определения с помощью команды *Clear["Global'*"]*.

Для получения наиболее полной информации о каком-либо объекте, встроенным в пакет *Mathematica*, необходимо с помощью мыши щелкнуть иконку *Help* в основном меню и выбрать раздел *Help*. В этом случае открывается *Help*-окно, в котором можно получить информацию обо всех встроенных функциях и обо всех имеющихся внешних программах, а также прочитать любой раздел из описания пакета *Mathematica* и посмотреть множество демонстрационных программ.

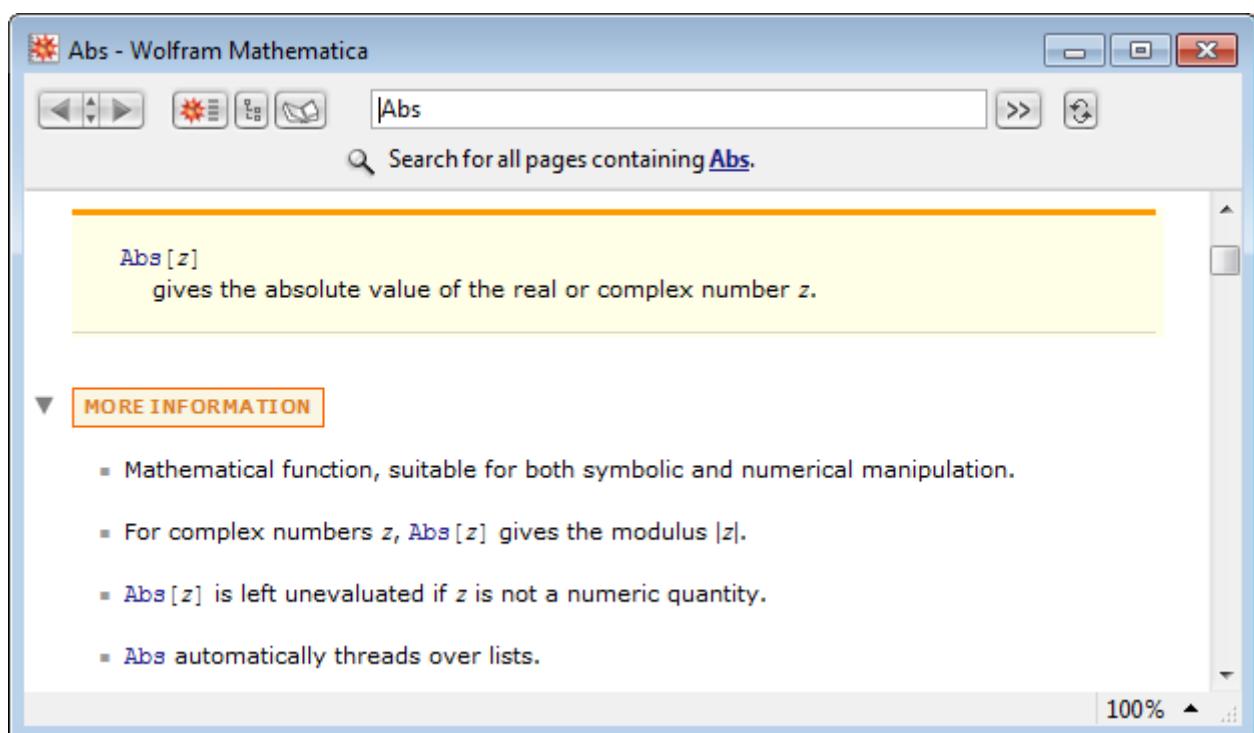
Получить информацию об объекте, который является встроенным в пакет *Mathematica* или введен пользователем, можно и следующим образом. Необходимо набрать знак вопроса «?» и затем имя этого объекта, например:



Для получения дополнительной информации об объекте необходимо набрать два знака вопроса без пробела и имя объекта



Также можно набрать имя встроенного объекта, выделить его и нажать клавишу *F1*, в открывшемся окне справки появится информация о выделенном объекте



Для окончания работы с пакетом *Mathematica* достаточно с помощью мыши выбрать команду *Exit* в разделе *File* основного меню. Перед выходом появляется вопрос, следует ли сохранить введенные данные. В случае положительного ответа появляется дополнительное окно, в котором можно определить имя сохраняемого документа. Соответствующий файл будет иметь расширение «.nb». При следующем сеансе работы с программой *Mathematica* этот файл может быть открыт с помощью команды *Open* в разделе *File* для дальнейшей работы с ним.

Основные операции и функции

Mathematica обеспечивает выполнение стандартных арифметических операций: +, -, /, *, ^, !. Причем произведение может быть представлено пробелом, то есть 2^*x и $2_{[\text{пробел}]}x$ дадут в результате $2x$.

Имена всех встроенных объектов (констант, функций) начинаются с заглавной буквы. Аргументы указываются в квадратных скобках и разделяются запятой. Некоторые функции, например *Random*, не требуют аргумента, однако также вызываются с квадратными скобками *Random[]*.

Переменные обозначаются обычно маленькими буквами. В качестве имени переменной может использоваться комбинация символов.

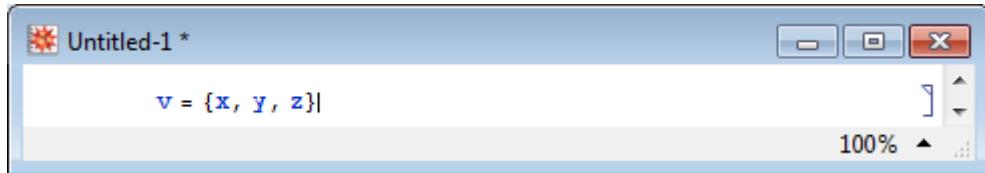
Для ввода объекта с палитры необходимо нажать на соответствующую кнопку. Однако некоторые объекты, имеющиеся на палитре, можно ввести с клавиатуры, нажав определенную комбинацию клавиш. Приведем некоторые из них в следующей таблице:

Операция	Комбинация клавиши	Результат
Степень	x [Ctrl+6] 3	x^3
Дробь	x [Ctrl+/] 3	$\frac{x}{3}$
Корень квадратный	[Ctrl+2] x	\sqrt{x}
Радикал	[Ctrl+2] x [Ctrl+5] 3	$\sqrt[3]{x}$
Индекс	x [Ctrl+_] 1	x_1
Верхний индекс	x [Ctrl+] b	x^b
Подиндекс	x [Ctrl++] i	x_i
$\alpha, \beta, \pi, \dots$	[Esc] a [Esc] [Esc] b [Esc] [Esc] Pi [Esc]	$\alpha\beta\pi$
Дифференциал	[Esc] dd [Esc] x	dx
Интеграл	[Esc] Int [Esc] x [Ctrl+6] a [Ctrl+Пробел] [Esc] dd [Esc] x	$\int x^a dx$
Частная производная	[Esc] pd [Esc][Ctrl+_] x [Ctrl+Пробел] x^y	$\partial_x x^y$
Следовательно, равносильно	[Esc]=>[Esc] [Esc]<=>[Esc]	$\Rightarrow\Leftarrow$
Стрелки	[Esc] -> [Esc]	\rightarrow
Неравно	[Esc] != [Esc]	\neq
Нестрогие неравенства	[Esc] >= [Esc] [Esc] <= [Esc]	$\geqslant\leqslant$
Движение курсора за пределы структуры	[Ctrl+Пробел]	

Текст, заключенный в скобки (* Текст *), не обрабатывается и представляет собой комментарий. Пары символов «(*» и «*)» набираются без пробела.

Векторы, матрицы, списки, таблицы

Для определения списков и матриц используются фигурные скобки: $\{\}$. Список или вектор представляет собой несколько объектов, заключенных в фигурные скобки и разделенных запятыми. Например:



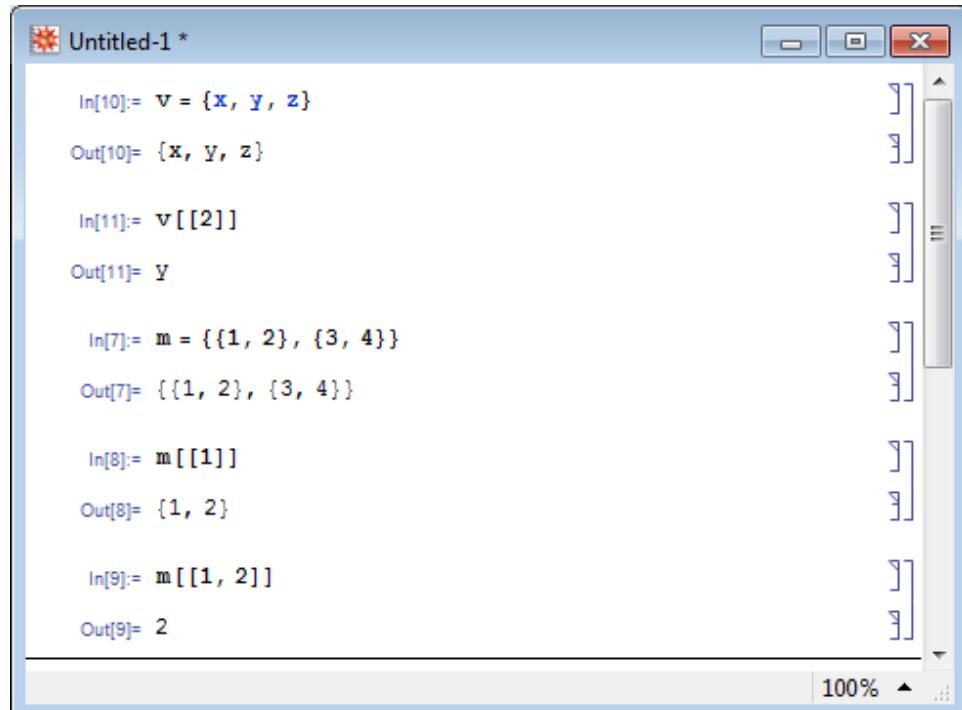
```
v = {x, y, z}
```

Матрица представляет собой вложенный список или список списков. Например:



```
m = {{1, 2}, {3, 4}}
```

Для выделения элементов списка используются двойные квадратные скобки. Для вектора v обозначение $v[[i]]$ соответствует i -му элементу списка. Отсчет элементов списка начинается с 1. Для матрицы m обозначение $m[[i,j]]$ соответствует i -ой строке матрицы, а $m[[i,j]]$ – j -му элементу i -ой строки



```
In[10]:= v = {x, y, z}
Out[10]= {x, y, z}

In[11]:= v[[2]]
Out[11]= y

In[7]:= m = {{1, 2}, {3, 4}}
Out[7]= {{1, 2}, {3, 4}}

In[8]:= m[[1]]
Out[8]= {1, 2}

In[9]:= m[[1, 2]]
Out[9]= 2
```

Для представления матрицы в обычной форме используется функция $MatrixForm[m]$ в префиксной записи или $m//MatrixForm$ в постфиксной записи

Альтернативным способом задания матрицы является выбор соответствующего объекта на палитре. Следует заметить, что с палитры можно вставить только матрицу размера 2×2 , при этом дополнительная строка добавляется нажатием комбинации $[Ctrl]+[Enter]$, а столбец – $[Ctrl]+[,]$.

Еще один способ задания векторов и матриц – это таблица *Table*. Так функция $Table[expr, \{i, min, max, шаг\}]$ создает список значений выражения $expr$ для каждого i . А функция $Table[expr, \{i, imin, imax, шаг\}, \{j, jmin, jmax, шаг\}]$ задает двумерный список $imax \times jmax$.

В пакет *Mathematica* встроены функции вычисления скалярного, векторного, поэлементного умножения векторов, вычисления определителя матрицы, ее собственных значений и векторов, транспонированной и обратной матрицы и др. Приведем для удобства некоторые из них в следующей таблице:

$\{\square, \square, \square, \square\}$	Непосредственное задание списка.
$Array[f, \{max\}]$	Задание списка вида $f[1], f[2], \dots, f[max]$
$Range[min, max, шаг]$	$\{min, min+шаг, \dots, max\}$
$v1.v2$	Скалярное умножение векторов $v1$ и $v2$
$v1*v2$	Поэлементное умножение векторов $v1$ и $v2$
$Cross[v1, v2]$	Векторное умножение векторов $v1$ и $v2$
$Det[\square]$	Определитель матрицы
$Eigenvalues[\square]$	Собственные значения матрицы
$Eigenvectors[\square]$	Собственные векторы матрицы
$Dot[A, B]$	Произведение матриц A и B
$Inverse[\square]$	Обратная матрица
$Transpose[\square]$	Транспонированная матрица
$Minors[\square]$	Список всех миноров матрицы

Выражения, функции

Пакет *Mathematica* позволяет производить различные символьные вычисления, например, преобразовывать алгебраические выражения. Для этого используется ряд встроенных функций. Некоторые из них имеют необязательные параметры, от указания или не указания которых зависит результат преобразования выражения. Приведем наиболее используемые функции в следующей таблице:

$Collect[\square, x]$	Представление в виде полинома по степеням x
$Coefficient[\square, form]$	Коэффициент при множителе $form$

Exponent[\square , form]	Максимальная степень множителя form
Factor[\square]	Разложение на множители
Factor[\square , Extension $\rightarrow\{\square\}$]	Разложение на множители с указанием алгебраического расширения
FactorTerms[\square, x]	Вынесение за скобки множителей, не зависящих от x
TrigFactor[\square]	Разложение на множители с использованием соотношений между тригонометрическими функциями
Expand[\square]	Раскрытие скобок
ComplexExpand[\square]	Раскрытие скобок в предположении, что все переменные являются действительными
PowerExpand[\square]	Раскрытие скобок с преобразованием $(xy)^n$ в $x^n y^n$
TrigExpand[\square]	Замена тригонометрических функций от кратных углов на степени и произведения тригонометрических функций и раскрытие скобок
Simplify[\square]	Упрощение выражений
Simplify[\square, \square]	Упрощение выражений с условием
FullSimplify[\square]	Полное упрощение (работает со спец. функциями)
Together[\square]	Приведение к общему знаменателю
TrigToExp[\square]	Запись тригонометрической функции через экспоненту
ExpToTrig[\square]	Запись экспоненты от комплексного аргумента через тригонометрические функции

Приведем несколько примеров преобразования выражений. Разложим на множители выражение $(1+x+x^3+x^4)$, а затем опять раскроем скобки

```

Untitled-1 *
In[1]:= Factor[1 + x + x^3 + x^4]
Out[1]= (1 + x)^2 (1 - x + x^2)

In[2]:= Expand [(1 + x)^2 (1 - x + x^2)]
Out[2]= 1 + x + x^3 + x^4

```

Применим функцию *Expand[]* к одному и тому же выражению с дополнительным условием и без него и сравним результаты

```

Untitled-1 *
In[6]:= Expand[(1 + x)^3 + (2 + x)^4 + (1 + y)^2]
Out[6]= 18 + 35 x + 27 x^2 + 9 x^3 + x^4 + 2 y + y^2

In[5]:= Expand[(1 + x)^3 + (2 + x)^4 + (1 + y)^2, 1 + x]
Out[5]= 1 + 3 x + 3 x^2 + x^3 + (2 + x)^4 + (1 + y)^2

```

Раскрываются скобки только у выражения вида $1+x$

Иногда, из-за недостатка информации, функция не преобразовывает выражение, а лишь переписывает его в выходную ячейку. Покажем это на примере работы функции *Simplify[]*, с условием и без него:

```
In[7]:= Simplify[Sqrt[z^2]]
Out[7]:= Sqrt[z^2]

In[9]:= Simplify[Sqrt[z^2], z < 0]
Out[9]:= -z
```

Представим выражение $(y^2+3y+x)(x^4-4x^3y^2+x^2y-2xy^3+8xy+1)$ в виде полинома по степеням x с помощью функции *Collect[]*, а затем выделим коэффициент при y^2 функцией *Coefficient[]*:

```
In[3]:= Collect[(x^4 - 4 x^3 y^2 + x^2 y - 2 x y^3 + 8 x y + 1) (y^2 + 3 y + x), x]
Out[3]:= x^5 + 3 y + y^2 + x^4 (3 y - 3 y^2) + x^2 (8 y + 3 y^2 - y^3) +
          x^3 (y - 12 y^3 - 4 y^4) + x (1 + 24 y^2 + 8 y^3 - 6 y^4 - 2 y^5)

In[4]:= Coefficient[%, y^2]
Out[4]:= 1 + 24 x + 3 x^2 - 3 x^4
```

Рассмотрим преобразования тригонометрических выражений на примерах раскрытия скобок в произведении тригонометрических функций, а также перехода от тригонометрической функции к экспоненте и обратно:

```
In[11]:= TrigExpand[Cos[2 z] Cos[z]]
Out[11]:= (Cos[z]^2 + Cos[z]^3)/2 - (3/2) Cos[z] Sin[z]^2

In[14]:= TrigToExp[4 Cos[2 x]]
Out[14]:= 2 e^-2 i x + 2 e^2 i x

In[15]:= ExpToTrig[e^i x]
Out[15]:= Cos[x] + i Sin[x]
```

Переходить от одной формы записи алгебраического выражения к другой можно с помощью оператора подстановки «/.», определяя правила замены любых символов в выражении на числа или другие символы и выражения. При этом само выражение не изменяется. Приведем примеры:

```

In[24]:= v = x + y + 2
Out[24]= 2 + x + y

In[19]:= v /. x -> z^2
Out[19]= 2 + y + z^2

In[21]:= v /. {x -> 1, y -> -2}
Out[21]= 1

In[22]:= v
Out[22]= 2 + x + y

```

Mathematica может производить вычисление сумм с помощью функции *Sum* или записи их в символьной форме со знаком « \sum », который добавляется с палитры. Количество слагаемых может быть как конечным, так и бесконечным. Например,

```

In[16]:= Sum[x^k/k!, {k, 0, 5}]
Out[16]= 1 + x + x^2/2 + x^3/6 + x^4/24 + x^5/120

In[18]:= \!\(\sum _{k=0}^5\frac{x^k}{k!}\)
Out[18]= 1 + x + x^2/2 + x^3/6 + x^4/24 + x^5/120

In[17]:= Sum[1/(2 i + 1), {i, 10}]
Out[17]= 17181176/14549535

```

Второй аргумент у функции *Sum*, определяющий интервал изменения переменной, по которой производится суммирование, может быть задан в нескольких формах. Форма $\{i, imin, imax, step\}$ задает изменение переменной i от значения $imin$ до $imax$ с шагом $step$. Если $step$ не указан, он по умолчанию равен 1. Форма $\{i, n\}$ задает изменение переменной i от 1 до n с шагом 1.

Если результат суммирования в символьной форме найти не удается, то *Mathematica* возвращает сумму не вычисленной. Однако для числовых рядов можно получить численную аппроксимацию суммы, используя функцию *NSum*.

Для бесконечных числовых рядов результат получается только в том случае, если ряд сходится:

```
In[20]:=  $\sum_{n=1}^{\infty} \frac{n}{n! + 1}$ 
Out[20]=  $\sum_{n=1}^{\infty} \frac{n}{1 + n!}$ 
In[21]:= NSum[n/(n! + 1), {n, 1, Infinity}]
Out[21]= 1.8065
```

Функция *Sum* позволяет выполнять суммирование по нескольким переменным. Так, *Sum*[*f*,{*i*,*imin*,*imax*},{*j*,*jmin*,*jmax*}] производит суммирование по переменным *i* и *j*, причем сумма по *i* является внешней:

```
In[23]:= Sum[x^i y^j, {i, 1, 3}, {j, 1, 2}]
Out[23]= x y + x^2 y + x^3 y + x y^2 + x^2 y^2 + x^3 y^2
```

Для вычисления произведений используется соответствующая символьная запись со знаком « \prod » или функции *Product* и *NProduct*, формат обращения к которым аналогичен формату функций *Sum* и *NSum*:

```
In[24]:= Product[i, {i, n}]
Out[24]= n!
In[25]:=  $\prod_{i=1}^n i$ 
Out[25]= n!
In[31]:= NProduct[3/(1+i), {i, 0, 5}]
Out[31]= 1.0125
```

Отметим также тот факт, что любое выражение в системе *Mathematica* представляет собой структуру списка, а следовательно, к любой его части можно обратиться как к элементу списка. Например, выделим из уравнения $a+b(x+1)=c+3x+4y$, второе слагаемоеевой части и первый множитель третьего слагаемого в правой части:

```

In[97]:= eq = a + b (1 + x) == c + 3 x + 4 y
Out[97]= a + b (1 + x) == c + 3 x + 4 y

In[98]:= eq[[1, 2]]
Out[98]= b (1 + x)

In[99]:= eq[[2, 3, 1]]
Out[99]= 4

```

Хотя в пакете *Mathematica* имеется большое количество встроенных функций, позволяющих производить различные вычисления, иногда требуется ввести новую функцию. Для этого нужно в левой части определения функции указать ее заголовок или имя, за которым в квадратных скобках следуют аргументы разделяемые запятыми, а в правой – выражение, определяющее функциональную зависимость $f[x_1, x_2, \dots, x_n] = F[x_1, x_2, \dots, x_n]$. Символ « $_$ » в левой части означает, что вместо x_1, x_2, \dots, x_n в качестве аргументов функции могут быть указаны любые выражения или числа

```

In[32]:= f[x_] = x^2 + 5 x + 1
Out[32]= 1 + 5 x + x^2

In[33]:= f[a]
Out[33]= 1 + 5 a + a^2

In[38]:= u[x_, y_] = Cos[x]^2 + Sin[y]
Out[38]= Cos[x]^2 + Sin[y]

In[40]:= u[\frac{\pi}{4}, \frac{\pi}{6}]
Out[40]= 1

```

При определении функции вместо оператора присвоения « $=$ » может быть использован оператор отложенного присвоения « $:=$ ». В этом случае f присваивается лишь правило ее вычисления, а само вычисление будет происходить лишь при обращении к этой функции:

```

In[41]:= f[x_] := x2 + 5 x + 1
In[42]:= f[a]
Out[42]= 1 + 5 a + a2
In[43]:= f[5]
Out[43]= 51

```

Для удаления значения, присвоенного функции, используется команда `Clear[]` или «`=.`»:

```

In[10]:= f[x_] := x2 + 5 x + 1
In[11]:= f[a]
Out[11]= 1 + 5 a + a2
In[12]:= Clear[f]
In[13]:= f[a]
Out[13]= f[a]

```

Задачи для самостоятельного решения

A. Задать матрицу $A(3 \times 3)$ с помощью палитры, а матрицу $B(3 \times 3)$ – с помощью функции `Table`. Вычислить сумму, разность произведение матриц, их определители, обратные и транспонированные матрицы, собственные значения и собственные векторы. Взяв любые два собственных вектора, найти результат их поэлементного, скалярного и векторного умножения:

$$1. A = \begin{pmatrix} 1 & 2 & 2 \\ 3 & 1 & -1 \\ 2 & 3 & 1 \end{pmatrix}, B : \{b_{ij} = i + j, \forall i, j = 1, 2, 3\}$$

$$2. A = \begin{pmatrix} -1 & -6 & 2 \\ 13 & 10 & -12 \\ 20 & -3 & 1 \end{pmatrix}, B : \{b_{ij} = ij, \forall i, j = 1, 2, 3\}$$

$$3. A = \begin{pmatrix} 1 & 2 & -2 \\ 11 & -1 & 1 \\ 14 & 23 & -8 \end{pmatrix}, B : \{b_{ij} = 2i - j, \forall i, j = 1, 2, 3\}$$

$$4. A = \begin{pmatrix} 8 & 7 & -5 \\ 3 & -5 & -1 \\ 2 & 10 & -21 \end{pmatrix}, B : \{b_{ij} = 3i + 2j, \forall i, j = 1, 2, 3\}$$

$$5. A = \begin{pmatrix} 2 & -12 & 2 \\ -3 & 2 & -3 \\ 2 & 3 & 2 \end{pmatrix}, B : \{b_{ij} = i - 2j, \forall i, j = 1, 2, 3\}$$

В. Привести подобные слагаемые в многочлене (a), разложить на множители выражение (b), раскрыть скобки в выражении (c), представить его в виде многочлена по степеням x и найти коэффициент при y^3 :

1.

- (a) : $x^5 - 8x^2 + x^4 + 12x - 8 + 21x^5 + 256x - 168x^3 + 31x^3$
- (b) : $x^4 - 8x^2 + 4x + 3$
- (c) : $x^2y - xy^2 + 2x^4y^4 + 12xy - 8y + 16 + 21x^3y^5 + 26y - 18y^3 + 31xy^3$

2.

- (a) : $12x^5 - 8x^3 + 5x^2 + 12x - 8 + 21x^3 + 26x - 128x^3 + 31x + 131x^2$
- (b) : $x^5 - 9x^4 + 20x^3 + 8x^2 - 37x - 15$
- (c) : $8x^3y^2 - 13xy^2 - x^2y^4 + 12x^3y - 8y + 16x + 21x^3y^5 + 26 - 18y^3$

3.

- (a) : $2x^6 - 8x^3 + 15x^4 + 12x - 8 + 121x^2 + 26x - 12x^3 + 3x - 11x^2$
- (b) : $x^4 + 2x^3 - 16x^2 - 2x + 15$
- (c) : $2x^2y^2 - 3xy^2 - x^2y^4 + 12x^3y - 18xy + 16x + 2x^3y^4 + 6 - 18xy^3$

4.

- (a) : $8x^3 + 15x^2 + 12x - 16 + 21x^3 + 26x^4 - 18x^3 + 31x + 31x^2 - 3x^4$
- (b) : $x^4 - 6x^3 - 19x^2 + 84x + 180$
- (c) : $x^3y^2 - 13xy^2 - 4x^2y^4 + 2x^2y - 18y + 16x + 21x^3y^3 + 6xy - 18y^3$

5.

- (a) : $12x^5 - 8x^3 + 5x^6 + 12x - 8 + 6x^5 - 128x^3 + 31x + x^2$
- (b) : $x^5 + 7x^4 - 20x^3 - 240x^2 - 576x - 432$
- (c) : $3x^3y^2 - 23x^2y^2 - 6x^2y^4 + 12x^3y - 8xy + 16x^3y^3 + 6 - 18y^3$

С. Определить функции f и g для вычисления суммы и произведения соответственно. Вычислить их для $n=1, 5, 10, m$:

$$1. \sum_{k=1}^n \frac{1}{k}, \prod_{k=1}^n \frac{1}{(k+1)(k+2)}$$

2. $\sum_{k=1}^n \frac{k-1}{k!}, \prod_{k=1}^n \frac{k}{(k+1)(k+2)}$
3. $\sum_{k=1}^n \frac{2k}{k+2}, \prod_{k=1}^n \frac{2k}{2k-1}$
4. $\sum_{k=1}^n \frac{2k}{2k-1}, \prod_{k=1}^n \frac{(-1)^k}{(k+1)(k+2)}$
5. $\sum_{k=1}^n \frac{(-1)^k}{k}, \prod_{k=1}^n \frac{2k-1}{(k+1)(k+2)}$

§2. РЕШЕНИЕ УРАВНЕНИЙ

Уравнение в системе *Mathematica* формируется двойным знаком равенства «==». В пакете существует шесть функций для решения уравнений в символьной форме и две функции для численного решения. Приведем их для удобства в следующей таблице:

<code>Solve[$\square == \square$, var]</code>	Решение уравнения относительно переменной var.
<code>Solve[$\{\square == \square, \square == \square\}$, {var1, var2}]</code>	Решение системы уравнений относительно var1 и var2.
<code>Roots[$\square == \square$, var]</code>	Решение уравнения относительно переменной var.
<code>Eliminate[$\{\text{syst}\}, \{\text{vars}\}$]</code>	Упрощение системы путем исключения неизвестных.
<code>Reduce[$\square == \square$, var]</code>	Решение уравнений с параметрами.
<code>SolveAlways[$\square == \square$, var]</code>	Нахождение условия, при котором уравнение переходит в тождество, относительно переменной var.
<code>LinearSolve[matrsyst, vectorrightparts]</code>	Решение системы линейных уравнений в матричной форме
<code>NSolve[$\square == \square$, var]</code>	Численное решение полиномиальных уравнений.
<code>FindRoot[$\square == \square$, {var, startpoint}]</code> <code>FindRoot[$\square == \square$, {var, firstpoint, secondpoint}]</code>	Численное решение уравнения.

Рассмотрим работу этих функций на примерах. Функция *Solve* позволяет находить символьные решения полиномиальных уравнений и некоторых неполиномиальных уравнений. Она имеет два обязательных аргумента `Solve[$\square == \square$, var]`, где первым аргументом является уравнение (список уравнений), а вторым – искомая переменная (список переменных):

In[32]:= `Solve[a x2 + b x + c == 0, x]`

Out[32]= $\left\{ \left\{ x \rightarrow \frac{-b - \sqrt{b^2 - 4 a c}}{2 a} \right\}, \left\{ x \rightarrow \frac{-b + \sqrt{b^2 - 4 a c}}{2 a} \right\} \right\}$

In[33]:= `Solve[e2-y2 == e-y, y]`

Solve::ifun :
Inverse functions are being used by Solve, so some solutions may not
be found; use Reduce for complete solution information. >>

Out[33]= $\left\{ \left\{ y \rightarrow -1 \right\}, \left\{ y \rightarrow 2 \right\} \right\}$

Предупреждение о том,
что могут быть найдены
не все решения.

Решения, получаемые с помощью *Solve*, представляют собой список правил замены, то есть выражений вида $x \rightarrow -4.5$, которые позволяют в любом выражении вместо x подставить соответствующее значение. Для обращения к любому из правил замены, как к элементу списка, используются двойные квадратные скобки `[[n]]`:

In[1]:= `Solve[x2 - 2 x + 3 == 0, x]`

Out[1]= $\left\{ \left\{ x \rightarrow 1 - i \sqrt{2} \right\}, \left\{ x \rightarrow 1 + i \sqrt{2} \right\} \right\}$

In[2]:= `%[[1]]`

Out[2]= $x \rightarrow 1 - i \sqrt{2}$

In[3]:= `x2 - 2 x + 3 == 0 /. %[[1]]`

Out[3]= `True`

Используется
первое правило
замены и
проверяется
справедливость

Для системы уравнений в качестве решения будем иметь список списков правил замены. В случае кратных корней данные элементы дублируются согласно величине их кратности:

In[1]:= `Solve[{1 + x2 == 0, y2 == x2}, {x, y}]`

Out[1]= $\left\{ \left\{ y \rightarrow -i, x \rightarrow -i \right\}, \left\{ y \rightarrow -i, x \rightarrow i \right\}, \left\{ y \rightarrow i, x \rightarrow -i \right\}, \left\{ y \rightarrow i, x \rightarrow i \right\} \right\}$

Для полиномиальных уравнений выше четвертого порядка невозможно записать решение в аналитической форме, поэтому для представления корней таких уравнений *Mathematica* использует *Root*-объект. Чтобы получить

численные значения найденных корней, достаточно применить к результату функцию N :

```
In[2]:= Solve[x^5 - 3 x + 1 == 0, x]
Out[2]= {{x → Root[1 - 3 #1 + #1^5 &, 1]}, {x → Root[1 - 3 #1 + #1^5 &, 2]}, {x → Root[1 - 3 #1 + #1^5 &, 3]}, {x → Root[1 - 3 #1 + #1^5 &, 4]}, {x → Root[1 - 3 #1 + #1^5 &, 5]}}
In[3]:= N[%]
Out[3]= {{x → -1.38879}, {x → 0.334734}, {x → 1.21465}, {x → -0.0802951 - 1.32836 i}, {x → -0.0802951 + 1.32836 i}}
```

Функция *Roots*, в отличие от *Solve*, выдает решение полиномиального уравнения в виде набора уравнений относительно искомой переменной. При этом уравнения разделяются двойной вертикальной чертой «||», которая обозначает логическое «или»:

```
In[5]:= Roots[2 x^2 + 4 x - 1 == 0, x]
Out[5]= x = 1/2 (-2 - Sqrt[6]) || x = 1/2 (-2 + Sqrt[6])
```

Для реализации процедуры упрощения системы уравнений путем исключения неизвестных служит функция *Eliminate*, которая вызывается с двумя параметрами *Eliminate[{syst},{vars}]*, где *syst* – список уравнений системы, *avars* – переменная или список переменных, которые необходимо исключить. При этом число уравнений и неизвестных сокращается:

```
In[6]:= Eliminate[{x^2 + b x y + c y^2 == 0, x + y == 0}, x]
Out[6]= c y^2 == (-1 + b) y^2

In[12]:= Eliminate[{x^2 + y^2 - z^2 == 1, x + 3 z == 0, y + x + 2 z^2 == 0}, {x, y}]
Out[12]= 17 z^2 - 12 z^3 + 4 z^4 == 1
```

Для решения уравнений или систем уравнений с параметрами удобно использовать функцию *Reduce*. Результат получается в виде набора уравнений, который содержит все возможные решения, включая дополнительные условия на параметры:

```
In[13]:= Reduce[a x2 + b x + c == 0, x]
Out[13]= 
$$\left( a \neq 0 \&& \left( x = \frac{-b - \sqrt{b^2 - 4 a c}}{2 a} \right) \mid\mid x = \frac{-b + \sqrt{b^2 - 4 a c}}{2 a} \right) \mid\mid \left( a = 0 \&& b \neq 0 \&& x = -\frac{c}{b} \right) \mid\mid (c = 0 \&& b = 0 \&& a = 0)$$

```

Операция логического «или»

Операция логического «и»

Функция *SolveAlways*, также вызывается с двумя аргументами *SolveAlways[□==□,var]* и определяет значения параметров, при которых уравнения системы обращаются в тождества для любых значений переменных:

```
In[15]:= SolveAlways[x2 + b x + c == (x - x1) (x - x2), x]
Out[15]= {b → -x1 - x2, c → x1 x2}
```

Теорема

Функция *LinearSolve* вызывается с двумя параметрами *LinearSolve[matrsyst,vectorrightparts]*, где *matrsyst* – квадратная матрица системы линейных уравнений, *vectorrightparts* – вектор правой части, и находит решения систем линейных уравнений, заданных в матричной форме:

```
In[16]:= LinearSolve[{{1, 2, 1}, {2, 1, 1}, {1, 1, 1}}, {3, 1, 2}]
Out[16]= {-1, 1, 2}
```

Для численного решения уравнений используются две функции: *NSolve* и *FindRoot*. Обе эти функции позволяют находить корни как одного уравнения, так и системы уравнений.

Функция *NSolve* позволяет найти все корни полиномиального уравнения или системы полиномиальных уравнений, и имеет два обязательных аргумента *NSolve[□==□,var]*, где первым аргументом является уравнение (список уравнений), а вторым – искомая переменная (список переменных).

Решение получается в виде списка правил замены. Для системы уравнений в качестве решения будем иметь список списков правил замены:

```

In[20]:= NSolve[x^4 - 2 x + 3 == 0, x]
Out[20]= {{x → -0.951325 - 1.19608 i}, {x → -0.951325 + 1.19608 i},
          {x → 0.951325 - 0.615984 i}, {x → 0.951325 + 0.615984 i}]

In[19]:= NSolve[{x^2 + y^2 == 1, 2 x + 3 y == 4}, {x, y}]
Out[19]= {{x → 0.615385 - 0.399704 i, y → 0.923077 + 0.266469 i},
          {x → 0.615385 + 0.399704 i, y → 0.923077 - 0.266469 i}}

```

Для решения произвольных уравнений используется функция *FindRoot*, которая, однако, находит только один корень. Эта функция вызывается обычно со следующими аргументами: *FindRoot[$\square == \square$, {var, startpoint}]*, где первый аргумент – уравнение, *var* – искомая переменная, *startpoint* – начальное приближение к решению. При этом уравнение решается методом касательных. Если второй аргумент функции имеет вид *{var, firstpoint, secondpoint}*, то уравнение решается методом хорд:

```

In[5]:= FindRoot[Sin[x] + 2 e^x, {x, 0}]
Out[5]= {x → -0.921025}

In[6]:= FindRoot[{e^x == 3 y, y^2 == x}, {{x, 1}, {y, 1}}]
Out[6]= {x → 1.18237, y → 1.08737}

```

Mathematica стремится представлять числа как можно более точно и выдает результат в такой же форме, в какой были представлены введенные данные. Например, иррациональное число $\sqrt{2}$ можно представить в виде десятичной дроби, однако *Mathematica* делает это только тогда, когда поступает соответствующая команда, или когда число $\sqrt{2}$ заданно приближенно, то есть содержит десятичную точку:

```

In[19]:= Solve[x^2 - 2 == 0, x]
Out[19]= {{x → -\sqrt{2}}, {x → \sqrt{2}}}

In[20]:= NSolve[x^2 - 2 == 0, x]
Out[20]= {{x → -1.41421}, {x → 1.41421}}

```

По умолчанию *Mathematica* выдает приближенный результат с точностью шесть значащих цифр. Количество значащих цифр, используемых при вычислениях, зависит от типа компьютера и определяется с помощью функции

Precision. Для преобразования точного результата в приближенный служит функция N . Эта функция позволяет получить результат с любой точностью. В качестве примера вычислим число π с точностью 30 значащих цифр:

```
In[21]:= N[\pi]
Out[21]= 3.14159

In[22]:= N[\pi, 30]
Out[22]= 3.14159265358979323846264338328
```

Иногда при вычислениях требуется замена близких к нулю чисел точным нулем, для этого используется функция $Chop[x, pogr]$, где x – число, а $pogr$ – порядок погрешности. Если число меньше порядка погрешности, то оно становится точным нулем, в противном случае остается без изменения:

```
Chop[1. 10-4, 1. 10-5]
Out[24]= 0.0001

Chop[1. 10-4, 1. 10-3]
Out[28]= 0
```

Задачи для самостоятельного решения

А. Решить уравнения численно и аналитически. Сравнить результат:

1. $x^4 - \frac{7}{3}x^3 - \frac{31}{2}x^2 + 18 = 0$
2. $x^4 + \frac{19}{6}x^3 + \frac{7}{2}x^2 - \frac{10}{3}x - 2 = 0$
3. $x^5 + 4x^4 + 3x^3 + 2x^2 + 2x - 12 = 0$
4. $x^5 + x^4 - 9x^3 - x^2 + 20x - 12 = 0$
5. $3x^5 + 5x^4 - 23x^3 - 17x^2 + 44x - 12 = 0$

Б. Решить систему линейных уравнений с матрицей A и вектором правых частей B :

1. $A = \begin{pmatrix} 1 & 1 & 2 \\ 2 & -1 & 2 \\ 4 & 1 & 4 \end{pmatrix}, B = \begin{pmatrix} 1 \\ -4 \\ -2 \end{pmatrix}$

$$2. \quad A = \begin{pmatrix} 3 & 2 & 1 \\ 2 & 3 & 1 \\ 2 & 1 & 3 \end{pmatrix}, \quad B = \begin{pmatrix} 5 \\ 1 \\ 11 \end{pmatrix}$$

$$3. \quad A = \begin{pmatrix} 4 & -3 & 2 \\ 6 & -2 & 3 \\ 5 & -3 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} -4 \\ -1 \\ -3 \end{pmatrix}$$

$$4. \quad A = \begin{pmatrix} 5 & 2 & 3 \\ 2 & -2 & 5 \\ 3 & 4 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} -2 \\ 0 \\ -10 \end{pmatrix}$$

$$5. \quad A = \begin{pmatrix} 2 & -12 & 2 \\ -3 & 2 & -3 \\ 2 & 3 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} -2 \\ -13 \\ 13 \end{pmatrix}$$

§3. ДИФФЕРЕНЦИРОВАНИЕ И РЕШЕНИЕ ПРОСТЕЙШИХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ.

Дифференцирование

Mathematica позволяет вычислять в символьной форме производные всех стандартных математических функций, а также производные специальных функций. В пакете также встроены функции для решения дифференциальных уравнений. Приведем их в следующей таблице:

D[f, x]	Первая производная по x
D[f, {x, n}]	n-ая производная по x
D[f[x,y], {x, n1}, {y, n2}]	Частная производная n1-го порядка по x и n2 по y
f'[x]	Первая производная по x (второй вариант)
∂ _{xy} f[x,y]	Частная производная 2-го порядка
Dt[f]	Полный дифференциал функции f
DSolve[□==□, y, x]	Решение дифференциального уравнения (ДУ)
DSolve[{□==□, □==□}, {y1, y2}, x]	Решение системы ДУ
DSolve[{□==□, □==□}, y, {x1, x2}]	Решение ДУ в частных производных
NDSolve[{□==□, begindata} y, {x, xmin, xmax}]	Численное решение ДУ
NDSolve[{□==□, □==□, begindata}, {y1, y2}, {x, xmin, xmax}]	Численное решение системы ДУ

Для вычисления производной используется функция D , обращение к которой имеет вид: $D[f,x]$, где f – функция от переменной x или алгебраическое выражение, содержащее переменную x :

```
In[2]:= D[x^x, x]
Out[2]= x^x (1 + Log[x])

In[3]:= D[Sin[x]^n, x]
Out[3]= n Cos[x] Sin[x]^{-1+n}
```

При вычислении производной некоторой функции $f(x)$ нужно обязательно указывать ее аргумент, в противном случае результат получится неверный:

```
In[4]:= D[f, x]
Out[4]= 0

In[5]:= D[f[x], x]
Out[5]= f'[x]
```

Поскольку функция $f(x)$ неизвестна, *Mathematica* записывает производную в символьной форме.

При вычислении n -й производной функция D вызывается в следующей форме: $D[f[x], \{x, n\}]$:

```
D[x^n + 5 x^{n-2} - x^2 - 1, {x, 2}]
Out[6]= -2 x + 5 (-2 + n) x^{-3+n} + n x^{-1+n}

D[x^n, {x, 5}]
Out[7]= (-4 + n) (-3 + n) (-2 + n) (-1 + n) n x^{-5+n}
```

Формат обращения к функции D при вычислении смешанной производной имеет вид: $D[f[x,y,z], \{x,n1\}, \{y,n2\}, \{z,n3\}]$, где $n1, n2, n3$ – порядки производных по переменным x, y, z соответственно:

```

In[8]:= D[Sin[x y], {x, 2}, y]
Out[8]= -x y2 Cos[x y] - 2 y Sin[x y]

In[10]:= D[Sin[x y], {x, 2}, {y, 2}]
Out[10]= -4 x y Cos[x y] - 2 Sin[x y] + x2 y2 Sin[x y]

```

Альтернативным вариантом вычисления производной является запись ее в символьной форме $f'[x]$, $f''[x]$ и так далее. Для частной производной $\partial_x f[x,y]$ –

```

In[16]:= y[x_]:= xn;
          y'''[x]
Out[17]= (-1 + n) n x-2 + n

In[15]:= \partial_x Sin[x y]
Out[15]= y Cos[x y]

```

Выражение $Dt[f]$ обозначает полный дифференциал от функции f . Так функция $Dt[f,x]$ вычисляет полную производную функции f по переменной x :

```

In[2]:= Dt[x3 + x y2, x]
Out[2]= 3 x2 + y2 + 2 x y Dt[y, x]

```

Здесь функция y не известна, и ее производная $Dt[y,x]$ не вычисляется.

Если в определении функции содержатся константы и требуется учесть равенство их производных нулю, то вводится соответствующая опция:

```

In[3]:= Dt[a x3 y + Sin[k x], x, Constants → {a, k}]
Out[3]= 3 a x2 y + k Cos[k x] + a x3 Dt[y, x, Constants → {a, k}]

```

Решение простейших дифференциальных уравнений

Для решения дифференциальных уравнений в аналитической форме в пакете *Mathematica* используется функция *DSolve*, формат обращения к которой имеет вид: $DSolve[\square == \square, y[x], x]$, где $\square == \square$ – дифференциальное уравнение

относительно функции $y(x)$. Функция y и все ее производные должны быть записаны с аргументом, заключенным в квадратные скобки: $y[x]$, $y'[x]$

```
In[1]:= DSolve[y'[x] == a y[x], y[x], x]
Out[1]= {{y[x] \rightarrow e^{ax} C[1]}}
```

Функция $DSolve$ стремится найти общее решение ДУ в явном виде и выдает результат в виде списка правил замены, причем каждое решение заключается в фигурные скобки. Для ДУ порядка n общее решение содержит n произвольных констант, которые обозначаются $C[1], C[2], \dots, C[n]$. Для получения частного решения необходимо в качестве первого аргумента $DSolve$ указать список, состоящий из самого уравнения и начальных или граничных условий:

```
In[6]:= DSolve[{y'[x] == a y[x], y[0] == 5}, y[x], x]
Out[6]= {{y[x] \rightarrow 5 e^{ax}}}
```

Найденные с помощью $DSolve$ решения можно подставить в любое выражение, содержащее $y(x)$. Однако это решение не определяет правил замены производных $y'(x), y''(x)$ и так далее, например:

```
In[7]:= y''[x] - 2 y'[x] + y[x] /. %[[1]]
Out[7]= 5 e^{ax} - 2 y'[x] + y''[x]
```

Чтобы получить решение, не имеющее этого недостатка, нужно в качестве второго аргумента функции $DSolve$ записать только имя искомой функции, не указывая ее аргумент. В этом случае решение представляется в виде чистой функции («purefunction»-объекта), в котором роль аргумента x , в некоторых случаях, играет символ «#1», а признаком этого объекта является символ «&». Полученное решение можно подставить в любое выражение, содержащее как функцию $y(x)$, так и ее производные:

```
In[16]:= DSolve[y'[x] == a y[x], y, x]
Out[16]= {{y \rightarrow Function[{x}, e^{ax} C[1]]}}
```



```
In[17]:= y''[x] - 2 y'[x] + y[x] /. %[[1]]
Out[17]= e^{ax} C[1] - 2 a e^{ax} C[1] + a^2 e^{ax} C[1]
```

Для решения систем уравнений в качестве первого аргумента функции указывается список уравнений, а в качестве второго аргумента – список искомых функций:

```
In[7]:= DSolve[{x'[t] == y[t], y'[t] == x[t]}, {x[t], y[t]}, t]
Out[7]= {{x[t] -> 1/2 e^-t (1 + e^2 t) C[1] + 1/2 e^-t (-1 + e^2 t) C[2],
y[t] -> 1/2 e^-t (-1 + e^2 t) C[1] + 1/2 e^-t (1 + e^2 t) C[2]}}
```

Если в список уравнений включить необходимое количество начальных или граничных условий, то будет найдено частное решение системы ДУ, не содержащее произвольных постоянных:

```
In[8]:= DSolve[{x'[t] == y[t], y'[t] == x[t], x[0] == 1, y[0] == 0}, {x[t], y[t]}, t]
Out[8]= {{x[t] -> 1/2 e^-t (1 + e^2 t), y[t] -> 1/2 e^-t (-1 + e^2 t)}}
```

Для некоторых уравнений решение может быть выражено через специальные функции, встроенные в пакет *Mathematica*. Если же *DSolve* не может найти аналитического решения ДУ, то *Mathematica* просто перепечатывает введенные данные в выходную ячейку:

```
In[3]:= DSolve[y'''[x] == x^2 / (Sqrt[9 - x^2] e^y[x] Cos[y[x]]), y'[x], y[x], x]
Out[3]= DSolve[y''[x] == e^-y[x] x^2 Sec[y[x]] y'[x] / Sqrt[9 - x^2], y[x], x]
```

В этом случае нужно преобразовать ДУ к более простому виду, используя правила, известные из теории дифференциальных уравнений. Если же аналитически решить уравнение не удается, можно попробовать решить его численно.

Для получения численного решения дифференциального уравнения используется функция *NDSolve*, которая вызывается по крайней мере с тремя аргументами: *NDSolve[$\{y' == f\}, \text{beginData}\}, y, \{x, xmin, xmax\}]$* , где $y' == f$ – дифференциальное уравнение относительно функции y , аргумент которой x изменяется в пределах отрезка $[xmin, xmax]$. Первый аргумент функции *NDSolve* представляет собой список, в который входит само дифференциальное уравнение и набор уравнений *beginData*, выражающих необходимое количество начальных или граничных условий. Так для дифференциального уравнения порядка n , необходимо записать n начальных условий. Функция y и ее производные, как и ранее, во всех уравнениях должны быть записаны с аргументом, заключенным в квадратные скобки: $y[x], y'[x]$:

```
In[14]:= sol =
NDSolve[{y''[x] - y'[x] + 3 x y[x] == 0,
y'[0] == 1, y[0] == 0}, y, {x, 0, 5}]

Out[14]= {{y \[Rule]
InterpolatingFunction[{{0., 5.}}, <>]}}
```

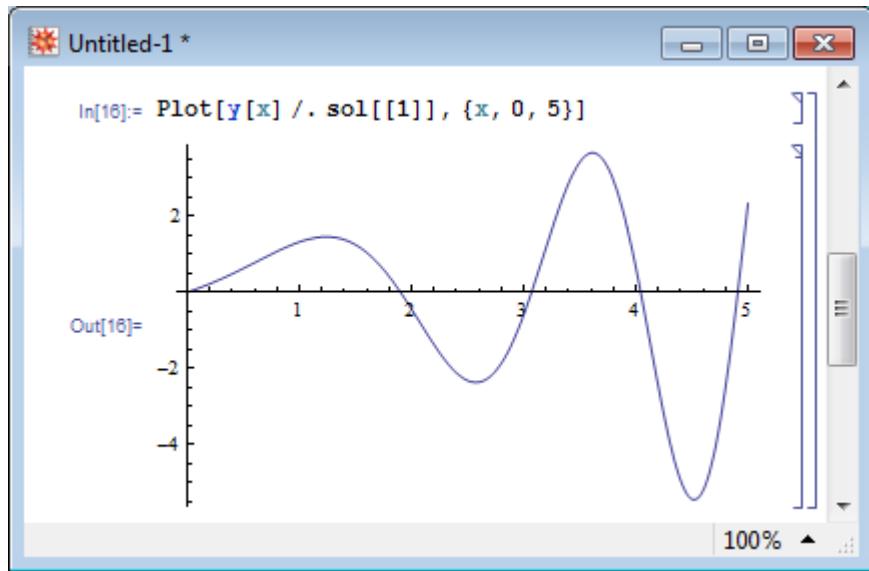
Обычно при выводе результата вычислений указывается только диапазон определения интерполяционной функции, а таблица данных изображается в виде символа « $<>$ ».

При использовании *NDSolve* искомая функция $y[x]$ находится в виде *InterpolatingFunction[$\{\{xmin, xmax\}\}, \langle\text{Таблица данных}\rangle$]* – объекта, который представляет собой таблицу значений функции $y(x_i)$ в различных точках из отрезка $[xmin, xmax]$ и позволяет найти значение функции в любой точке этого отрезка путем интерполяции табличных данных. При этом предполагается, что в промежутках между заданными точками функция является достаточно гладкой:

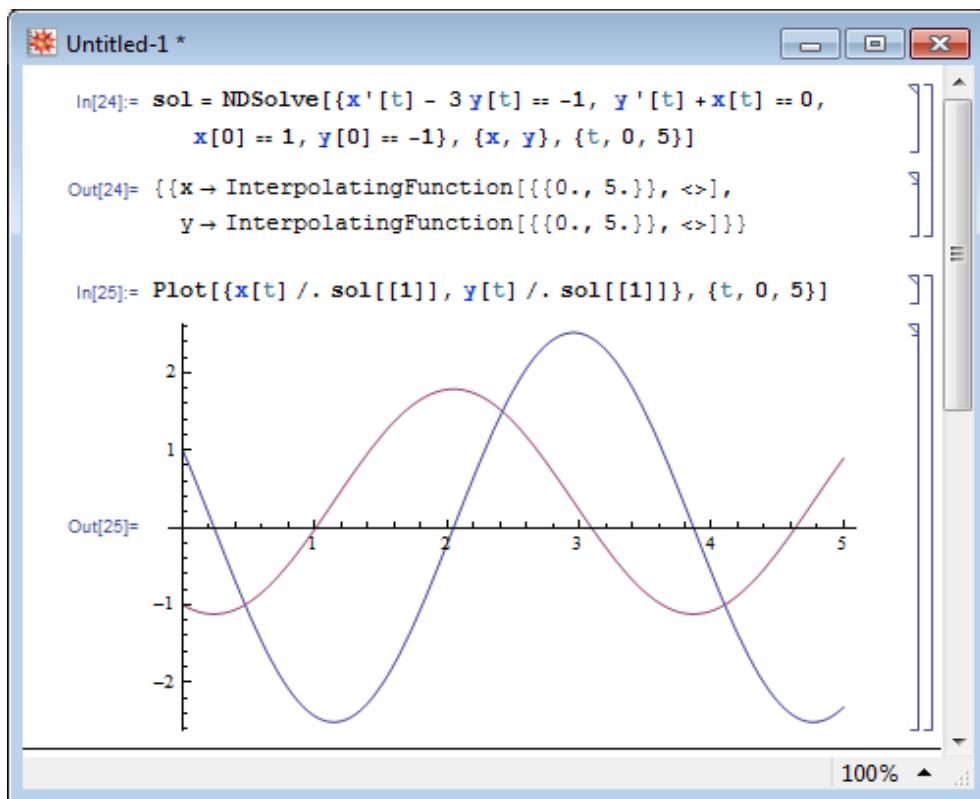
```
In[15]:= y[1.5] /. sol[[1]]

Out[15]= 1.25078
```

С интерполяционной функцией можно обращаться так же, как с любой встроенной функцией, то есть дифференцировать, интегрировать, строить ее график. Построим график полученного выше решения:



В случае решения системы дифференциальных уравнений в качестве первого аргумента используется список уравнений системы и начальные условия, а вторым аргументом является список искомых функций: $NDSolve[\{x' == \dots, y' == \dots, \dots, beginData\}, \{y1, y2, \dots\}, \{x, xmin, xmax\}]$:



§4. ИНТЕГРИРОВАНИЕ ФУНКЦИЙ И ВЫЧИСЛЕНИЕ ИНТЕГРАЛОВ

Для вычисления интегралов в пакете *Mathematica* имеется несколько встроенных функций. Приведем их для удобства в следующей таблице:

Integrate[f, x]	Интегрирование функции f по переменной x
-----------------	--

Integrate[f, x,y]	Интегрирование функции f по переменным x и y
Integrate[f, {x, xmin, xmax}]	Определенный интеграл
Integrate[f,{x, xmin, xmax},{y, ymin, ymax}]	Двойной определенный интеграл
Integrate[f, {x, xmin, xmax}, PrincipalValue->True]	Интегрирование с учетом главных значений
Integrate[f, {x, xmin, xmax}, Assumptions->{□}]	Интегрирование с условием на параметры
NIntegrate[f, {x, xmin, xmax}]	Численное вычисление определенного интеграла

Mathematica вычисляет большинство интегралов, которые выражаются через стандартные математические функции, а также позволяет интегрировать выражения, содержащие спецфункции. Для вычисления неопределенных интегралов $\int f(x)dx$ используется функция *Integrate[f,x]* или символьная запись $\int f(x)dx$, где символы « \int » и « d » набраны с палитры (или определенной комбинацией клавиш). Произвольная константа, которая возникает при интегрировании, полагается равной нулю:

```

Untitled-1 *

In[1]:= Integrate[Sin[x], x]
Out[1]= -Cos[x]

In[2]:= \int Sin[x] dx
Out[2]= -Cos[x]

```

Функция *Integrate* и запись с символом « \int » позволяет вычислять кратные неопределенные интегралы:

```

Untitled-1 *

In[3]:= Integrate[x Sin[x] Cos[y], x, y]
Out[3]= (-x Cos[x] + Sin[x]) Sin[y]

In[4]:= \int \int x Sin[x] Cos[y] dx dy
Out[4]= (-x Cos[x] + Sin[x]) Sin[y]

```

Для вычисления определенного интеграла в качестве второго аргумента функции *Integrate* необходимо использовать список трех элементов: $\{x,xmin,xmax\}$, где x – переменная интегрирования, а $xmin$ и $xmax$ – пределы интегрирования. Для символьной записи интеграла также можно указать пределы интегрирования:

```

In[5]:= Integrate[x Sin[x], {x, 0, \pi}]
Out[5]= \pi

In[6]:= \int_0^{\pi} x \sin[x] dx
Out[6]= \pi

```

С помощью функции *Integrate* можно вычислять и кратные определенные интегралы. Формат обращения к функции *Integrate* в этом случае имеет вид: *Integrate[f,{x, xmin, xmax},{y, ymin, ymax}]*. Первым указывается интервал изменения той переменной, интегрирование по которой производится в последнюю очередь:

```

In[11]:= Integrate[x Cos[x y], {x, 0, 1}, {y, 0, \pi}]
Out[11]= \frac{2}{\pi}

```

При интегрировании функцией *Integrate* можно накладывать условия на параметры:

```

In[15]:= Integrate[t^{a-1} (1-t)^{b-1}, {t, 0, 1}, Assumptions -> {Re[a] > 0, Re[b] > 0}]
Out[15]= \frac{\pi \csc [b \pi ] \Gamma [a]}{\Gamma [1-b] \Gamma [a+b]}

```

Эта функция позволяет также интегрировать в смысле главного значения (по Коши):

```

In[12]:= Integrate[\sqrt{x}/(x-1), {x, 0, 3}]
Integrate::idiv : Integral of \sqrt{x}/(-1+x) does not converge on {0, 3}. >
Out[12]= \int_0^3 \frac{\sqrt{x}}{-1+x} dx

In[14]:= Integrate[\sqrt{x}/(x-1), {x, 0, 3}, PrincipalValue -> True]
Out[14]= 2 \left(\sqrt{3}-\text{ArcCoth}\left[\sqrt{3}\right]\right)

```

Если *Mathematica* не может представить результат интегрирования в виде формулы, она выдает введенное выражение без изменений:

The screenshot shows a Mathematica notebook window titled "Untitled-1 *". In the input field (In[9]), the command `Integrate[1/(x + Cos[2 x]), {x, 0, π}]` is entered. In the output field (Out[9]), the result is displayed as an unevaluated integral expression: $\int_0^{\pi} \frac{1}{x + \cos[2x]} dx$.

Однако определенный интеграл, не содержащий неизвестных параметров, можно вычислить численно с помощью функции `NIntegrate[f[x], {x, xmin, xmax}]`.

Задачи для самостоятельного решения

A. Вычислить интегралы аналитически:

1. $\int \frac{\arcsine^x}{e^x} dx, \iint \frac{\sin x}{\cos y} dxdy$
2. $\int \frac{x^3 - 2x + 1}{e^x} dx, \iint \frac{x}{\cos y} dxdy$
3. $\int \frac{e^x}{x^5 + 1} dx, \iint \frac{\sin x}{2y^2 + 3y + 1} dxdy$
4. $\int \frac{\arcsin x}{\cos y} dx, \iint (x^3 + 2xy + y) ydxdy$
5. $\int \frac{x^3 + 2x + 1}{x^2 + x + 1} dx, \iint \frac{xy + x^2 + y^2}{y+1} dxdy$

B. Вычислить интегралы численно. Вычислить интегралы аналитически, для тех случаев, где это возможно, и сравнить результаты:

1. $\int_0^2 \sqrt{2}x^3 dx, \int_0^1 \int_{-x^2}^{\sqrt{x}} (9x^2y^2 + 48x^3y^3) dxdy$
2. $\int_1^e x\sqrt{1+x^2} dx, \int_0^1 \int_{-x^2}^{\sqrt[3]{x}} (18x^2y^2 + 32x^3y^3) dxdy$
3. $\int_0^{\pi} x\sqrt{\cos^2 3x + 4} dx, \int_0^1 \int_{-x^2}^x (12xy + 27x^2y^2) dxdy$
4. $\int_0^2 \frac{x^3(x+1)}{x-2} dx, \int_1^{2\sqrt{x}} \int_{\frac{1}{x}}^y y \ln x dxdy$
5. $\int_1^2 \frac{x^3 + 2x + 1}{e^x} dx, \int_0^1 \int_{-x^2}^x (2xy + 7x^2) dxdy$

§5. РАЗЛОЖЕНИЕ В РЯД И ВЫЧИСЛЕНИЕ ПРЕДЕЛОВ

Произвольную функцию $f(x)$ можно разложить в степенной ряд около точки $x=x_0$ с точностью до $(x-x_0)^n$, используя функцию *Series*, формат обращения к которой имеет вид: *Series*[$f[x]$, { x, x_0, n }]. Например:

```
In[1]:= Series[f[x], {x, x0, 3}]
Out[1]= f[x0] + f'[x0] (x - x0) +
  1/2 f''[x0] (x - x0)^2 + 1/6 f'''[x0] (x - x0)^3 + O[x - x0]^4
```

Полученное выражение представляет собой разложение функции $f(x)$ в ряд Тейлора в точке $x=x_0$, причем слагаемое вида $O[x-x_0]^4$ указывает, что следующий член ряда имеет порядок $(x-x_0)^4$. Функция *Series* генерирует также ряды, в которых содержатся рациональные и отрицательные степени $(x-x_0)$, отсутствующие в рядах Тейлора:

```
In[2]:= Series[Sin[x]/x^4, {x, 0, 4}]
Out[2]= 1/x^3 - 1/(6 x) + x/120 - x^3/5040 + O[x]^5

In[4]:= Series[Sin[Sqrt[x]], {x, 0, 4}]
Out[4]= Sqrt[x] - x^{3/2}/6 + x^{5/2}/120 - x^{7/2}/5040 + O[x]^{9/2}
```

Функция *Series* позволяет также производить разложение функции в окрестности бесконечно удаленной точки:

```
In[7]:= Series[1/Sin[x], {x, Infinity, 5}]
Out[7]= 1/x - 1/(6 x^3) + 1/(120 x^5) + O[1/x]^6
```

Если функция не может быть представлена в виде степенного ряда в окрестности некоторой точки (например, содержит сингулярность в этой точке), то *Mathematica* не производит разложения функции и выдает введенную функцию без изменения:

```

Untitled-1 *
In[2]:= Series[Cos[1/x], {x, 0, 5}]
Out[2]= Cos[1/x]

```

Слагаемое вида $O[x-x_0]^n$ в разложении, получаемом с помощью *Series*, является признаком того, что это выражение представляет собой аппроксимацию некоторой функции степенным рядом. Поэтому добавление к сумме произвольных функций вызывает преобразование всей суммы в степенной ряд с заданной точностью:

```

Untitled-1 *
In[5]:= Cos[x] + Sin[x] + O[x]^5
Out[5]= 1 + x - x^2/2 - x^3/6 + x^4/24 + O[x]^5

```

Над степенными рядами можно производить различные операции: сложение, умножение, возведение в степень, дифференцирование, интегрирование и другие, при этом получаются новые степенные ряды.

Для преобразования степенного ряда в обычное выражение используется функция *Normal*. При ее применении к степенному ряду происходит отбрасывание остаточного члена $O[x-x_0]^n$:

```

Untitled-1 *
In[6]:= Normal[1 + x - x^2/2 - x^3/6 + x^4/24 + O[x]^5]
Out[6]= 1 + x - x^2/2 - x^3/6 + x^4/24

```

С помощью функции *Series* можно также производить разложение выражений в ряд по нескольким переменным. Такая операция эквивалентна последовательному применению функции *Series* по переменным x и y в том порядке, в котором они стоят в качестве аргументов функции *Series* в исходном выражении:

Предел функции $f(x)$ при стремлении x к заданному значению x_0 используется функция $Limit$, которая вызывается в виде $Limit[f(x), x \rightarrow x_0]$:

Предел некоторых функций при стремлении x к x_0 зависит от направления, вдоль которого происходит приближение к x_0 . Соответствующее направление определяется опцией $Direction$, для которой указывается одно из трех значений: $Automatic$, 1 , -1 . Значения 1 и -1 определяют соответственно пределы снизу и сверху. Для значения $Direction \rightarrow Automatic$, которое используется по умолчанию, вычисляется предел сверху, кроме случая, когда $x \rightarrow Infinity$:

В случае, когда предельное значение функции не существует (например, функция является быстро осциллирующей), результатом вычисления соответствующего предела является $Interval$ -объект, который только указывает интервал, в котором может находиться вычисляемый предел:

```

Untitled-1 *

In[15]:= Limit[Cos[1/x], x -> 0]
Out[15]= Interval[{-1, 1}]

```

Задачи для самостоятельного решения

A. Найти представления в виде ряда для функций:

1. $f(x) = \frac{7}{12 + x - x^2}$
2. $f(x) = \ln(1 - x - 20x^2)$
3. $f(x) = \frac{\arctan x}{x}$
4. $f(x) = (3 + e^{-x})^2$
5. $f(x) = \frac{x^2}{\sqrt{4 - 5x}}$

B. Вычислить пределы:

1. $\lim_{n \rightarrow \infty} n(\sqrt[n]{x} - 1), (x > 0)$
2. $\lim_{n \rightarrow \infty} \sum_{k=1}^n \sin\left(\frac{ka}{n^2}\right)$
3. $\lim_{x \rightarrow 0} \frac{(1+mx)^n - (1+nx)^m}{x^2}$
4. $\lim_{x \rightarrow 1} \frac{x^m - 1}{x^n - 1}$
5. $\lim_{n \rightarrow \infty} \frac{4n^2 - \sqrt[4]{x^3}}{\sqrt{n^6 + n^3 + 1}}$

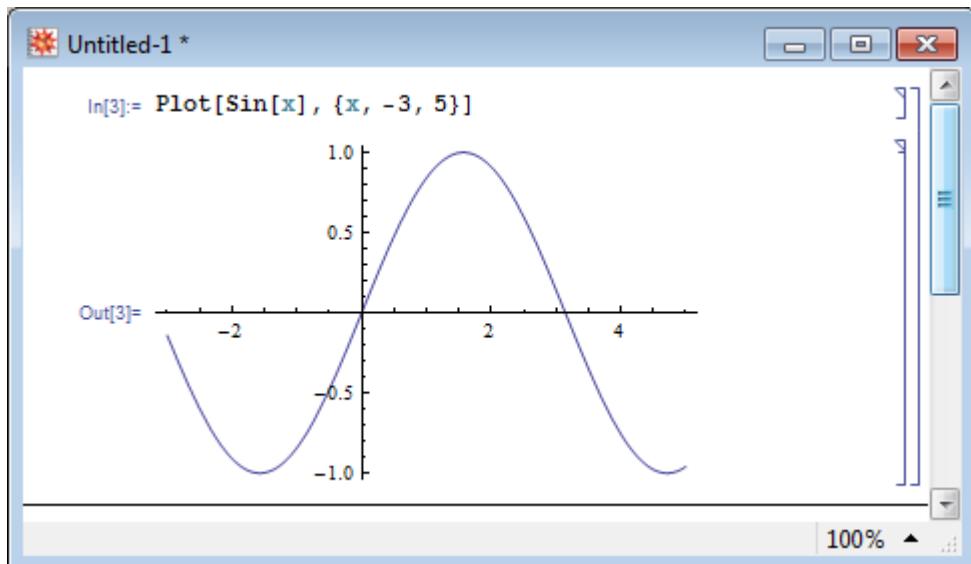
§6. ПОСТРОЕНИЕ ГРАФИКОВ ФУНКЦИЙ

Графики функций одного аргумента

Для построения отображения графиков функций одного аргумента в пакете *Mathematica* предусмотрено несколько функций. Приведем их для удобства в следующей таблице и рассмотрим каждую в отдельности.

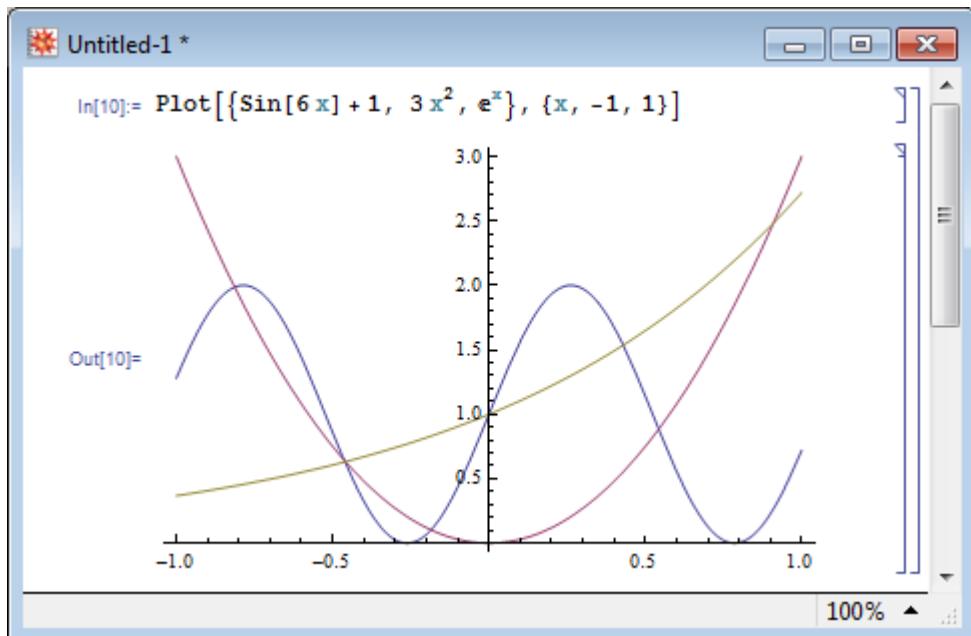
<code>Plot[f[x], {x, xmin, xmax}]</code>	График функции одного аргумента
<code>ParametricPlot[{fx[t], fy[t]}, {t, tmin, tmax}]</code>	График функции, заданной параметрически
<code>ListPlot[{□}]</code>	Графическое изображение списка чисел
<code>Show[g1,g2,...]</code>	Отображение графических объектов
<code>GraphicsArray[{g1,g2,...}]</code>	Отображение массива графических

Графики функций одного аргумента строятся с помощью функции *Plot*, формат обращения к которой имеет вид *Plot[f[x], {x, xmin, xmax}]*, где *f[x]* – заданная функция аргумента *x*, график которой строится на отрезке *[xmin,xmax]*. Следует отметить, что данная функция позволяет строить графики как конечных функций, так и функций, имеющих на заданном отрезке особенности. Построим на отрезке $[-3, 5]$ график функции $\sin(x)$:



Размеры графика можно менять. Для этого достаточно щелкнуть по графику левой кнопкой мыши и растянуть его за бегунки.

На одной координатной плоскости можно одновременно построить графики нескольких функций. Для этого в качестве первого аргумента функции *Plot* нужно указать список функций:

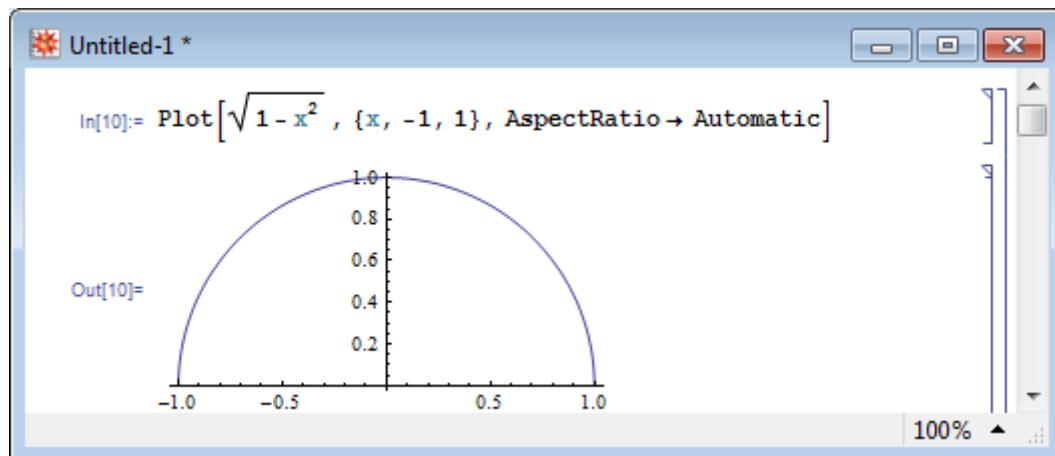


При построении графика система *Mathematica* должна определить в каких точках заданного отрезка следует вычислить значения функций, какую область изменения функции отобразить на графике и при каком масштабе, как провести координатные оси и другое. Управление этими процессами осуществляется

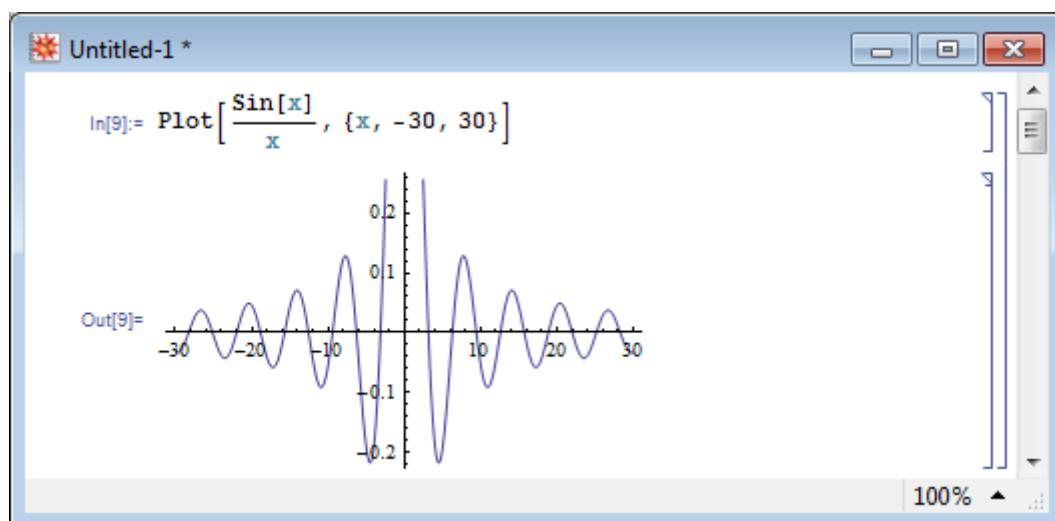
рядом опций, каждая из которых имеет свое имя и задается правилом замены вида $Option \rightarrow Value$. Список опций вместе с их значениями по умолчанию можно получить с помощью команды *Options*.

Если нужно изменить значение каких-либо опций, они указываются при вызове функции *Plot* в качестве третьего и последующих аргументов. Рассмотрим влияние некоторых функций на вид получаемого графика.

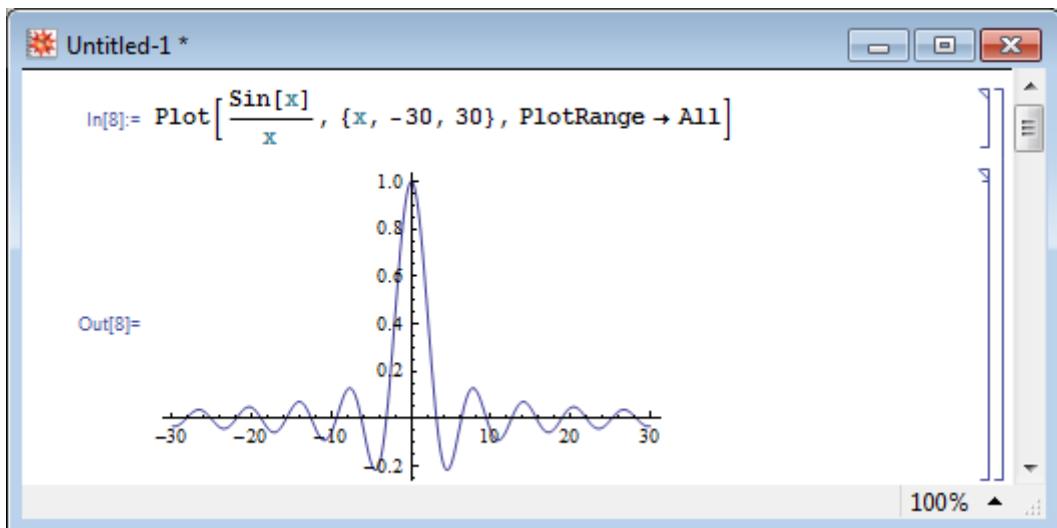
Обычно единичные отрезки на осях x и y имеют различную длину. Однако в некоторых случаях необходимо построить график таким образом, чтобы масштабы по осям абсцисс и ординат были одинаковыми. Это достигается выбором для опции *AspectRatio* значения *Automatic*:



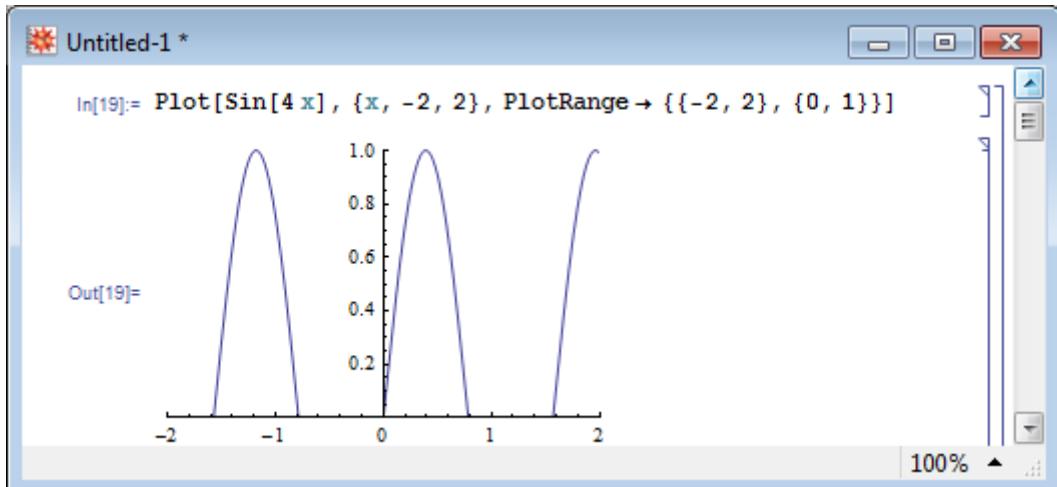
Если функция очень быстро возрастает или имеет особенности, то по умолчанию область больших значений функции не изображается, чтобы более детально показать ту область, где амплитуда изменений функции мала:



Для того, чтобы отобразить всю область изменения функции, нужно указать в качестве дополнительного параметра опцию *PlotRange* \rightarrow All:

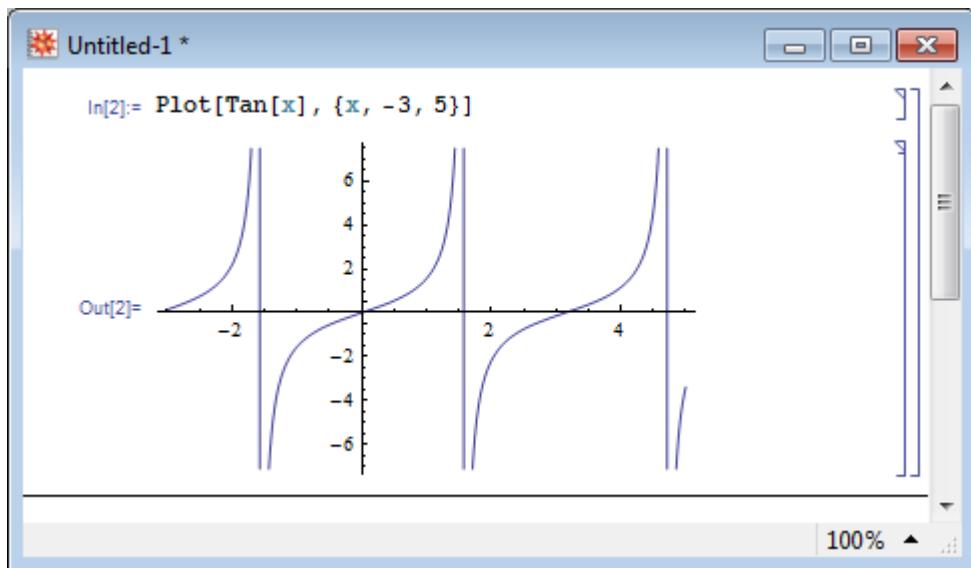


В качестве значения *PlotRange* можно также указать конкретный интервал изменения функции, который должен быть отображен на графике. Причем в первую очередь указывается интервал по оси абсцисс, а затем интервал по оси ординат:

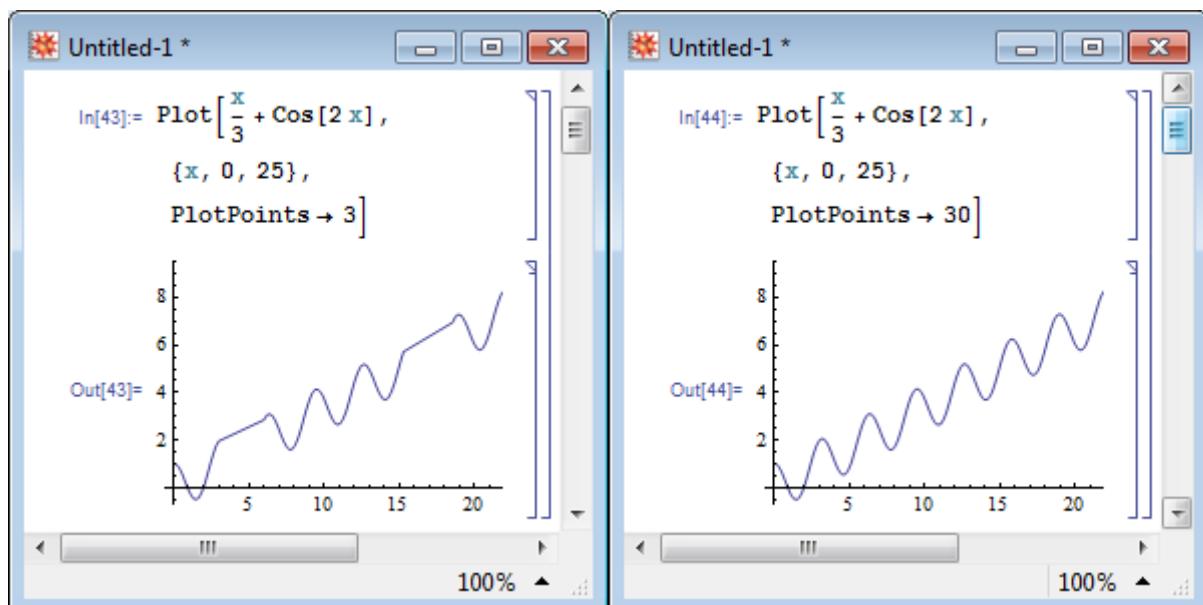


Размеры области на экране, в которой изображается график, определяются опцией *ImageSize* и задаются в пикселях, например *ImageSize -> {100, 100}*. При этом опция *PlotRegion* указывает, в какой части выделенной под график области, он должен располагаться, так значение $\{\{0, 1\}, \{0, 1\}\}$ указывает на всю выделенную область, а значение $\{\{0, 0.5\}, \{0.5, 1\}\}$ – ее левую верхнюю четверть.

При построении графика функции $f(x)$ на координатной плоскости отмечается множество точек $(x, f(x))$ для различных x из заданного интервала, и соседние точки соединяются отрезками прямых. Поскольку соседние точки соединяются отрезками прямой, то, при наличии у функции особенности на заданном интервале, на графике автоматически изображается соответствующая асимптота:

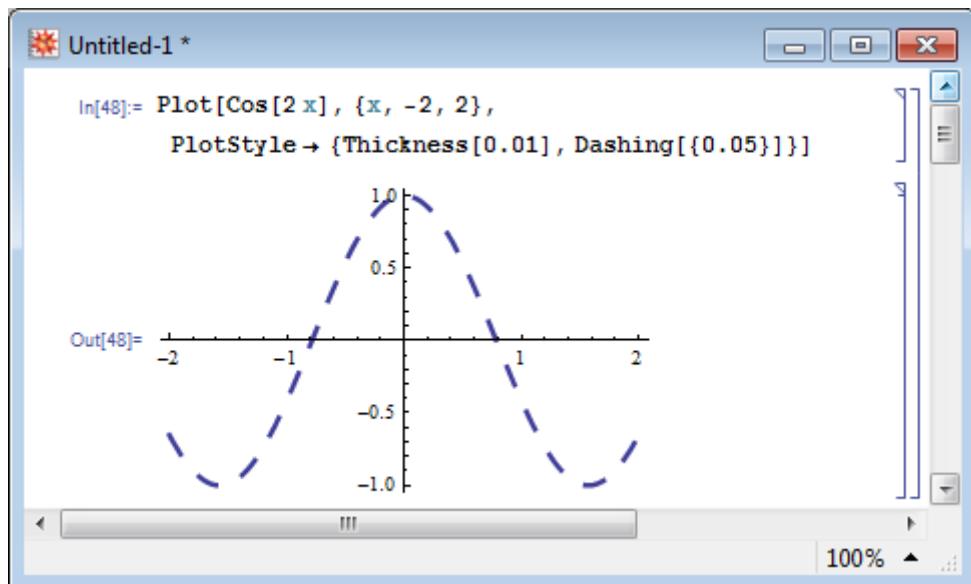


Для гладкости функции необходимо задать достаточное количество точек, причем на промежутках, где функция быстро изменяется, количество точек должно быть больше. Количество точек разбиения устанавливается опцией *PlotPoints*:

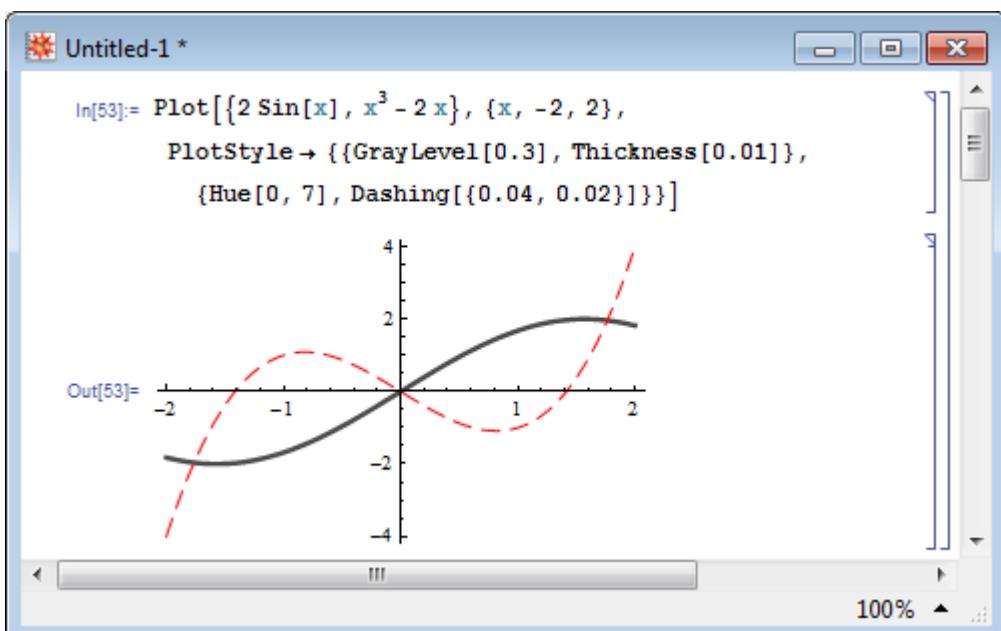


При построении графика происходит автоматическая компиляция функции, что может снижать точность вычислений. Если при построении графика необходима высокая точность, следует определить опцию *Method -> {Compiled -> False}*.

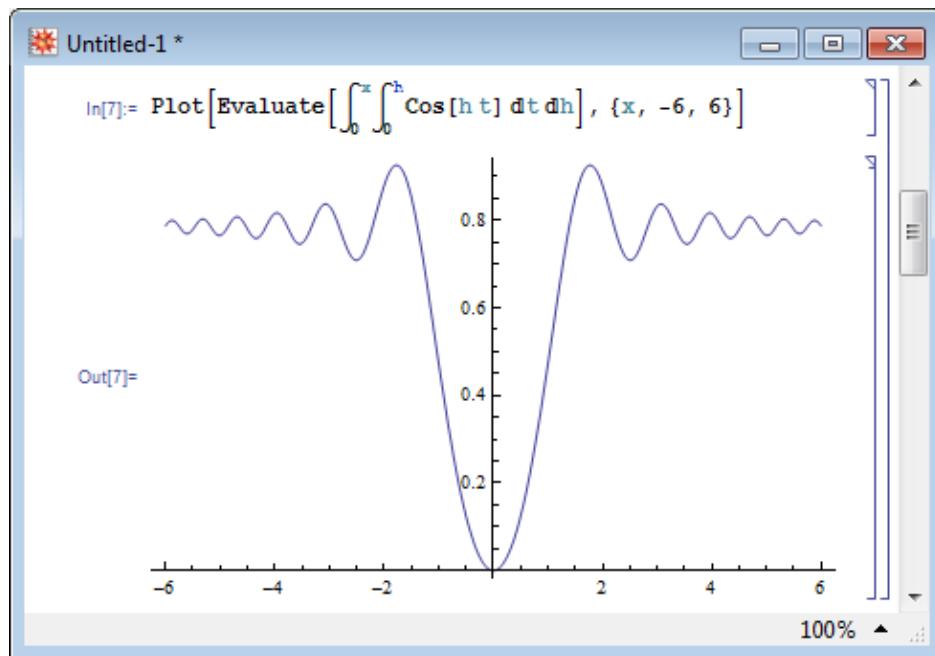
Опция *PlotStyle* позволяет задавать толщину, цвет и стиль кривой, изображаемой на графике, и представляет собой список директив. Толщина линии определяется директивой *Thickness*, аргумент которой – отношение ширины линии к ширине всего графика. Директива *AbsoluteThickness* определяет толщину линии в абсолютных единицах. Цвет линии определяется директивой *GrayLevel* (аргумент от 0 до 1), директивы *RGBColor[r,g,b]* (каждый аргумент от 0 до 1) или *Hue[h]* (аргумент от 0 до 1). Директива *Dashing[{d1,d2,...}]* позволяет изобразить линию пунктиром. Ее аргументы – длины изображаемых и неизображаемых сегментов кривой в относительных единицах:



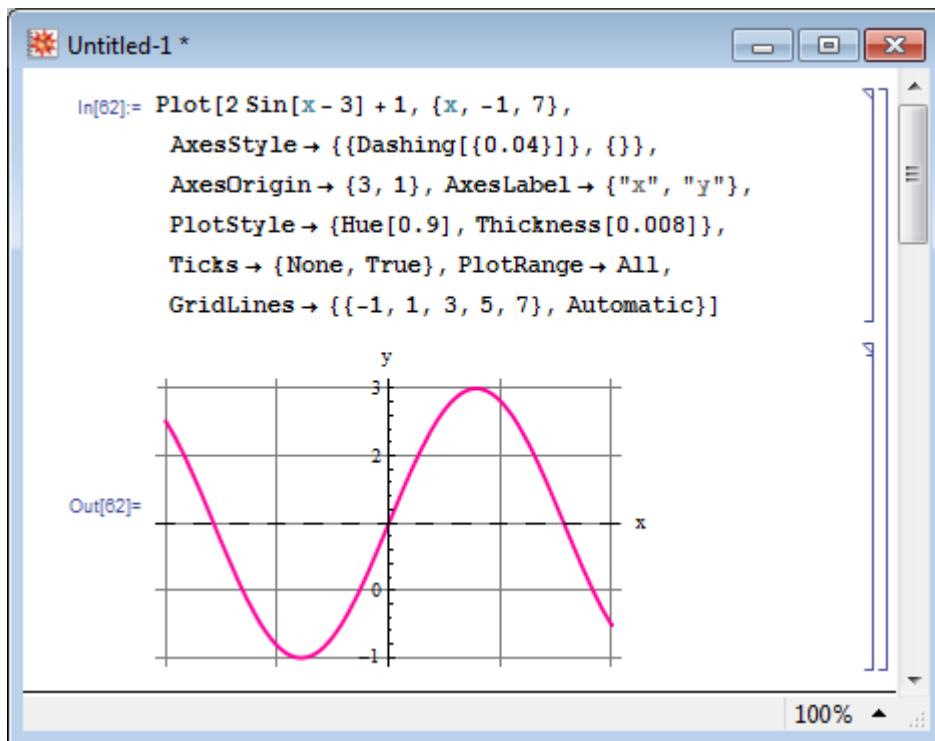
При построении нескольких кривых на одной координатной плоскости для каждой из них можно задать свой стиль:



Опция *Axes*→*True* указывает, что нужно изобразить оси. Если показывать оси не следует, нужно указать значение *False*. Опция *AxesStyle* определяет стиль изображения осей и может принимать те же значения, что и *PlotStyle*:

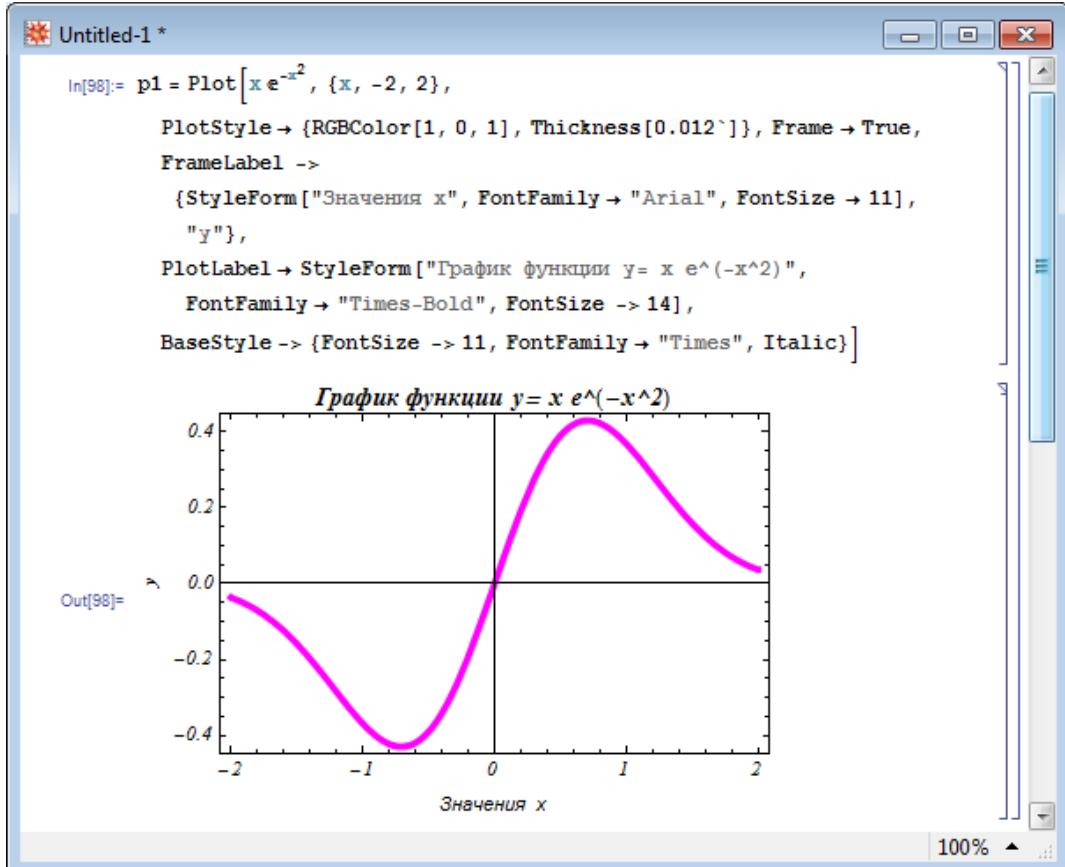


Опция *AxesOrigin* определяет точку пересечения осей. По умолчанию на каждой из осей наносятся метки, указывающие используемый масштаб. Если указать опции *Ticks→None*, то соответствующие метки на осях изображаться не будут. Опция *AxesLabel* позволяет ввести обозначение для каждой оси. Опция *GridLines* позволяет нанести на график координатную сетку:

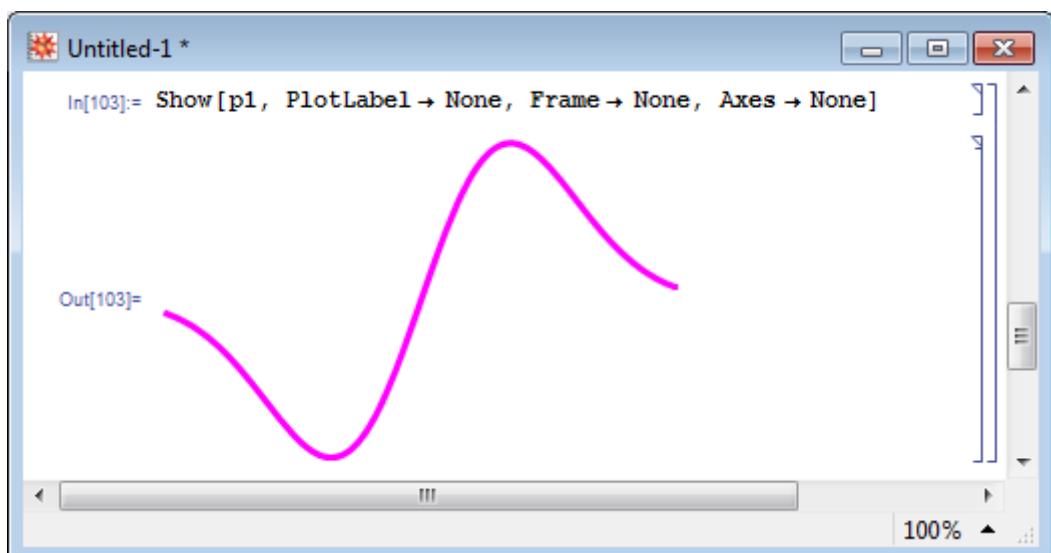


Опция *Frame* определяет наличие или отсутствие рамки вокруг графика. Стиль линии, используемой для рисования рамки, задается опцией *FrameStyle*. При наличии рамки метки с осей переносятся на стороны рамки, во избежание этого следует указать опцию *FrameTicks→None*. Опция *FrameLabel* позволяет сделать надписи вдоль сторон рамки, которые указываются в следующем порядке: снизу, слева, сверху, справа. Общий заголовок для графика можно нанести с помощью опции *PlotLabel*. По умолчанию для всех надписей и меток

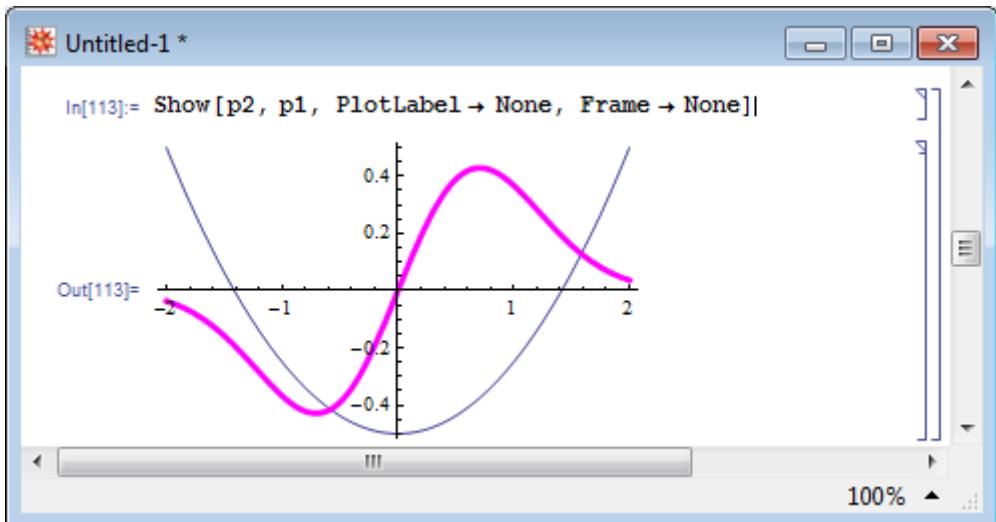
на осях используется шрифт, тип и размер которого определяется опцией *BaseStyle*. Если какую-либо надпись необходимо выполнить другим шрифтом, то можно использовать директиву *StyleForm* [*“Текст”, FontSize→размер, FontFamily→шрифт*]:



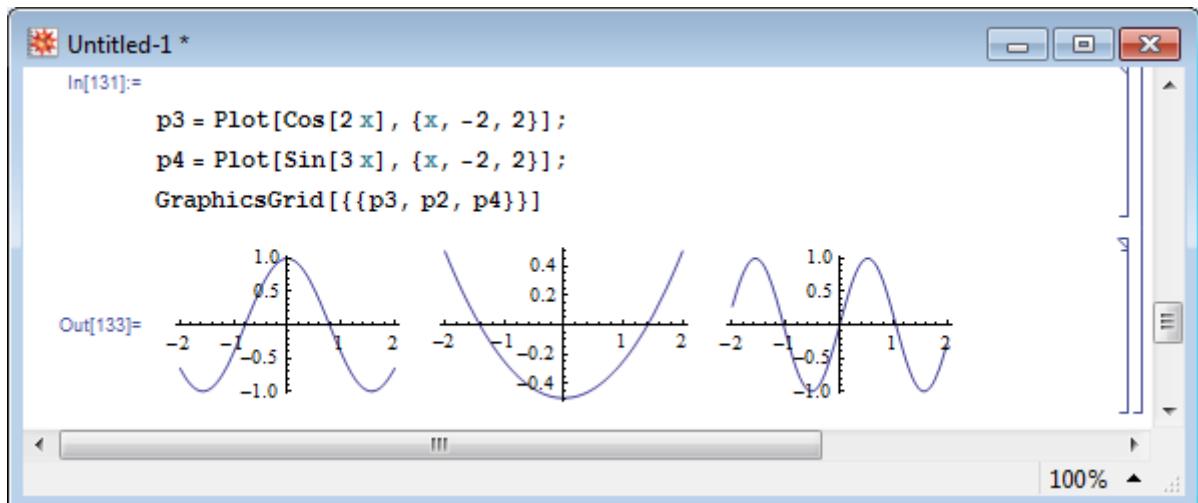
Если после функции *Plot* поставить «*;*», то график не будет выводиться на экран. Однако *Mathematica* сохраняет информацию обо всех построенных объектах, и с помощью функции *Show* можно вывести на экран любой построенный ранее график без повторных вычислений. Причем значение некоторых опций можно изменить:



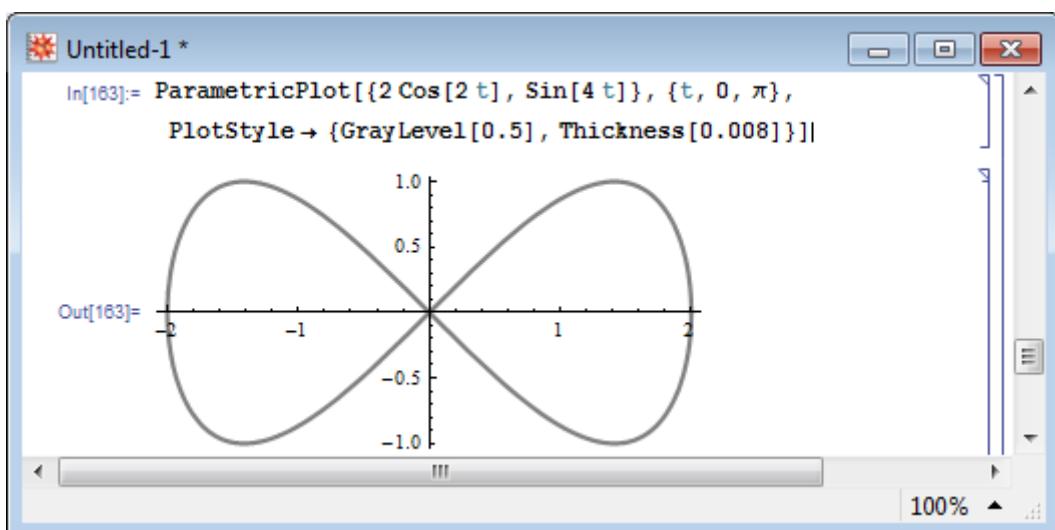
Функция *Show* позволяет также совместить несколько полученных ранее графиков на одной координатной плоскости:



Для отображения массива графических объектов используется функция *GraphicsGrid*[{*g*₁₁,*g*₁₂,...}, {*g*₂₁,*g*₂₂,...}, ...]:

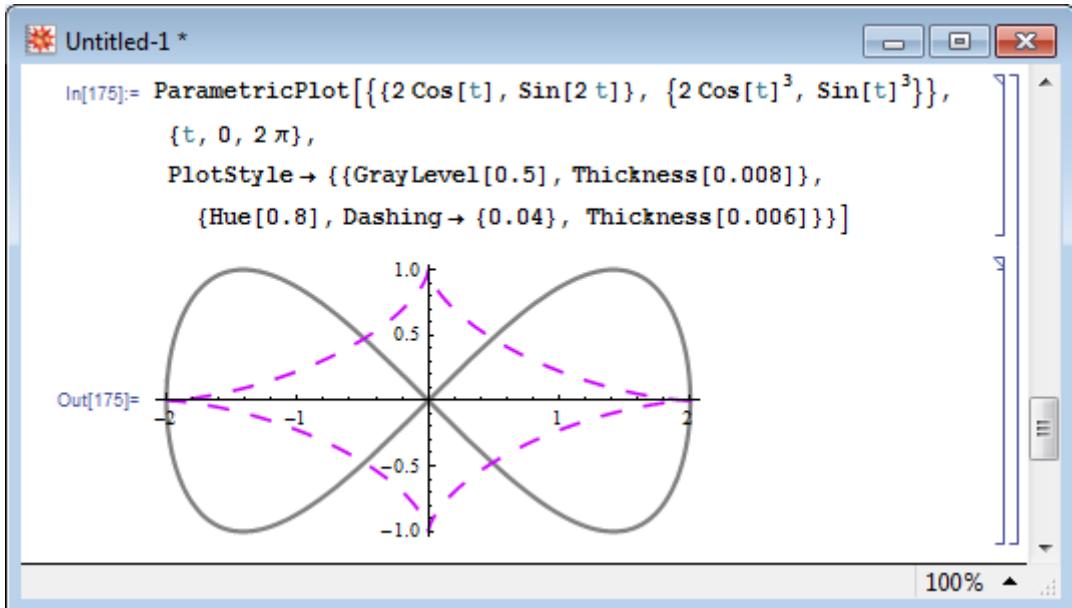


Для построения двумерного графика функции, заданной параметрически, используется функция *ParametricPlot*, формат обращения к которой имеет следующий вид: *ParametricPlot*[{*f*_x[*t*], *f*_y[*t*]}, {*t*, *t*_{min}, *t*_{max}}, *Option*→*Value*]:

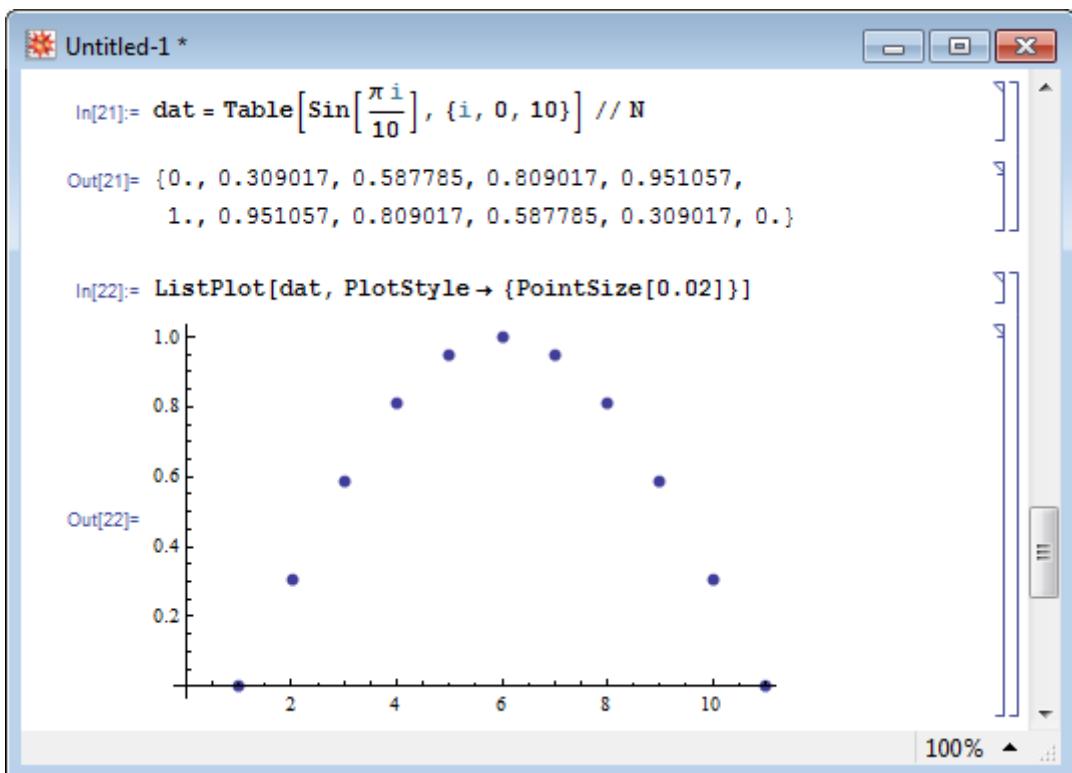


Эта функция имеет тот же набор опций, что и функция *Plot*, и позволяет строить график как одной, так и нескольких функций. При построении

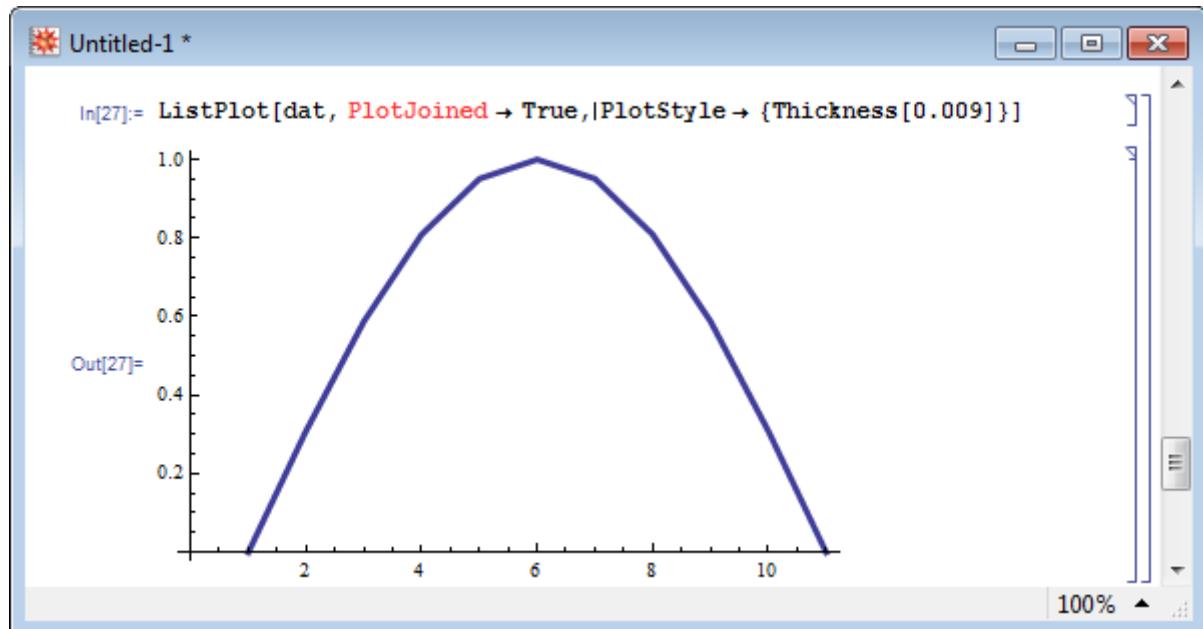
графиков нескольких функций в качестве первого аргумента функции *ParametricPlot* необходимо указать список функций, заданных параметрически:



В случае, когда требуется отобразить список чисел или функцию, заданную последовательностью численных значений вида $\{y_1, y_2, \dots, y_n\}$ или $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, используется функция *ListPlot*. Причем графики представляют собой множества точек с координатами (i, y_i) или (x_i, y_i) соответственно ($i=1, 2, \dots, n$). Зададим, например, список численных значений функции $y=\cos(i\pi/5)$ при $i=0, \dots, 10$ и визуализируем полученные данные:



Стиль изображения точек на графике определяется опцией *PlotStyle*. Директива *PointSize* отвечает за размер точек на графике, а директива *Thickness* – за толщину линии, если точки соединяются. Соединять точки на графике или нет, указывает опция *PlotJoined*:



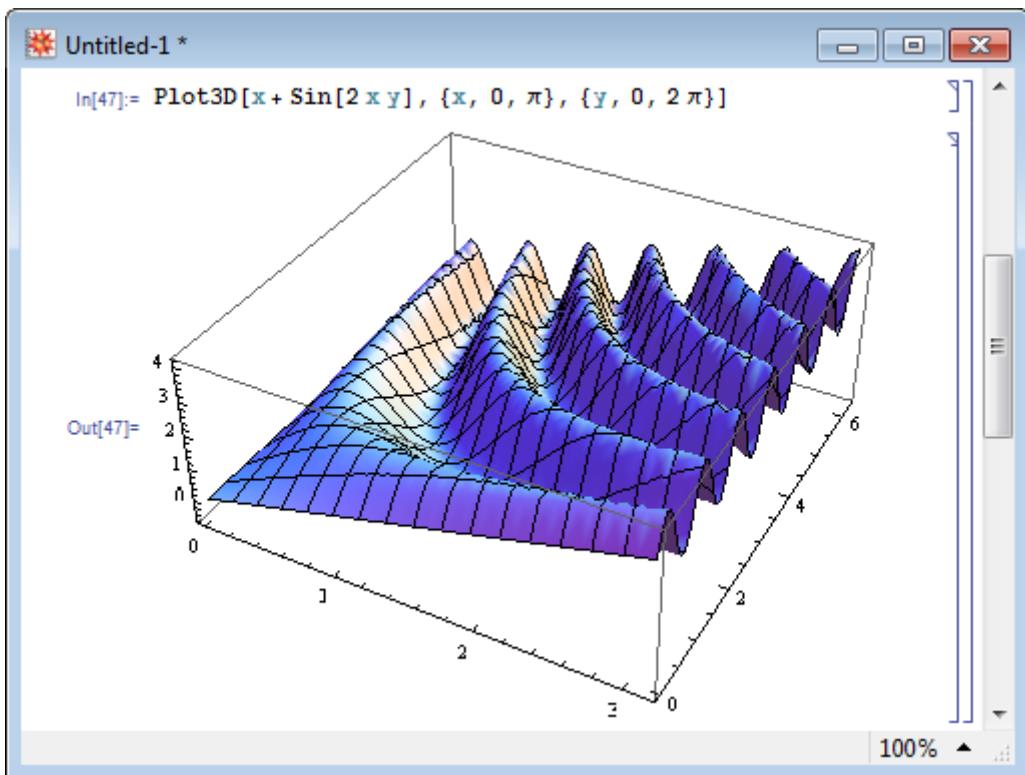
Остальные опции *ListPlot* аналогичны соответствующим опциям функции *Plot*.

Визуализация функций двух аргументов

Для построения функций двух аргументов в системе *Mathematica* существует несколько возможностей. Приведем их в следующей таблице и рассмотрим каждую в отдельности подробно.

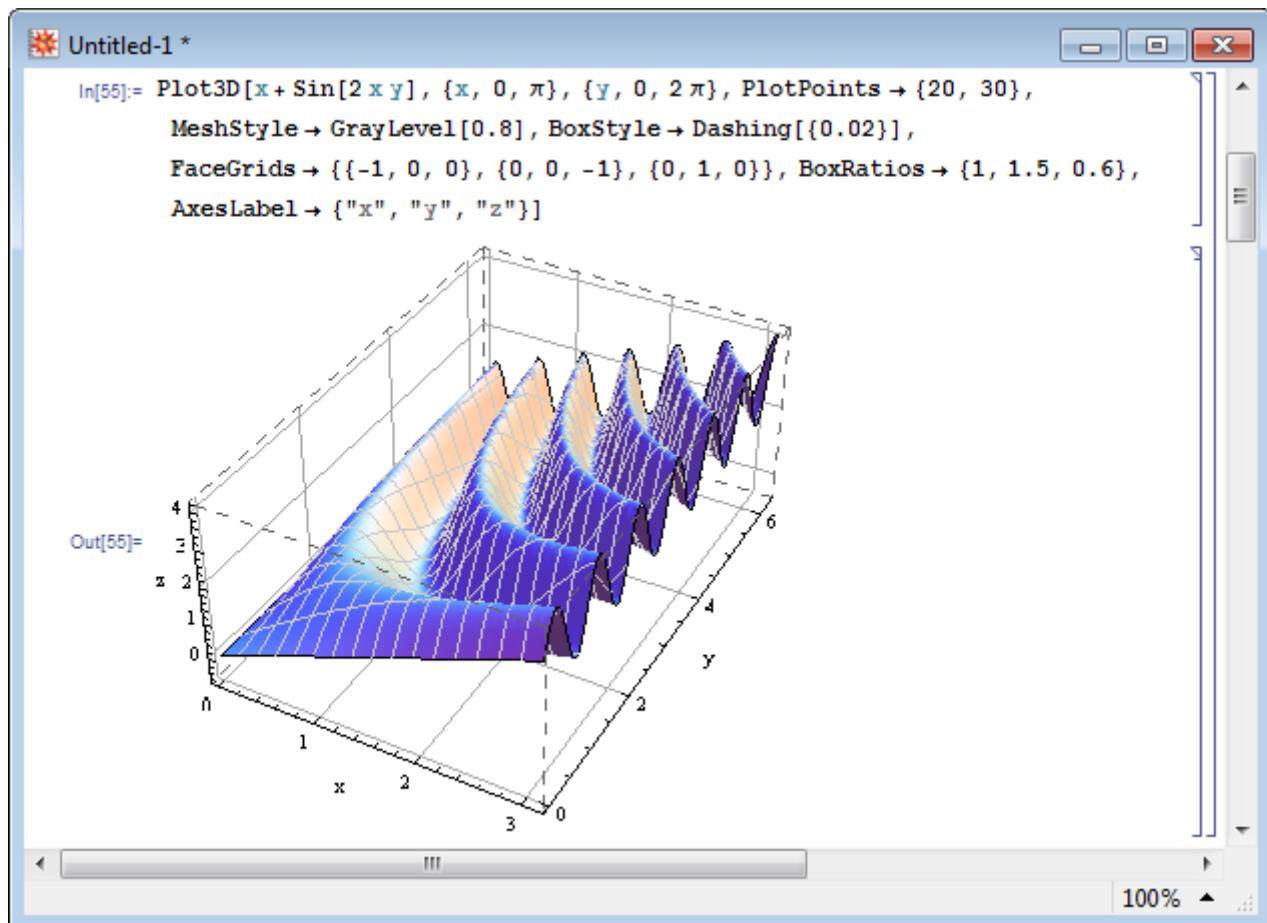
<code>Plot3D[f[x,y], {x, xmin, xmax}, {y, ymin, ymax}]</code>	Трехмерный график
<code>ParametricPlot3D[{fx[u,v],fy[u,v],fz[u,v]}, {u, umin, umax}, {v, vmin, vmax}]</code>	График параметрически заданной функции
<code>ContourPlot[f[x,y],{x,xmin,xmax},{y,ymin,max}]</code>	Изображение линий уровня
<code>DensityPlot[f[x,y],{x,xmin,xmax},{y,ymin,max}]</code>	График функции плотности
<code>ListPlot3D[{{□,□},...}]</code>	Изображение списка
<code>ListContourPlot[{{□,□},...}]</code>	Изображение линий уровня функции, заданной списком значений
<code>ListDensityPlot[{{□,□},...}]</code>	График функции плотности, заданной списком значений

Функция двух аргументов $f(x,y)$ определяет в трехмерном пространстве некоторую поверхность. Для ее визуализации можно использовать функцию *Plot3D*, формат обращения к которой имеет вид *Plot3D[f[x,y], {x, xmin, xmax}, {y, ymin, ymax}]*, где $f[x,y]$ – заданная функция аргументов x и y , которые изменяются соответственно в пределах отрезков $\{x, \text{xmin}, \text{xmax}\}$ и $\{y, \text{ymin}, \text{ymax}\}$. Построим, например, поверхность, определяемую функцией $f=x+\sin(xy)$:

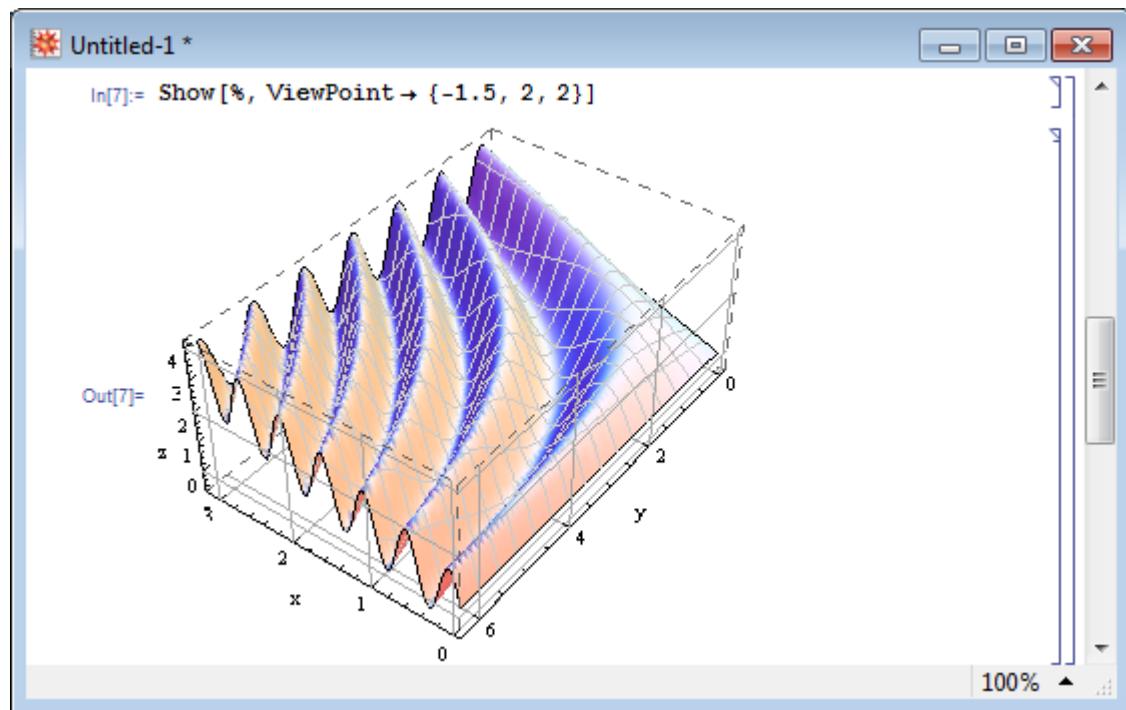


Вид получаемой поверхности определяется значениями ряда опций, часть из которых (*AspectRatio*, *Axes*, *AxesLabel*, *AxesStyle*, *Background*, *BaseStyle*, *ColorOutput*, *Compiled*, *DefaultColor*, *DisplayFunction*, *Epilog*, *PlotLabel*, *PlotPoints*, *PlotRange*, *PlotRegion*, *Prolog*, *Ticks*) совпадают с соответствующими опциями функции *Plot*. Полный список опций выводится по команде *Options*. Опции, как и при построении функций одной переменной, указываются в качестве четвертого и последующих аргументов.

Рассмотрим некоторые особенности построения трехмерных графиков. Для построения более детального изображения имеет смысл увеличивать значение опции *PlotPoints*, причем количество точек по осям *x* и *y* может быть различным, например, *PlotPoints* $\rightarrow\{20,30\}. Чтобы легче было представить пространственную структуру поверхности, по умолчанию на нее наносится прямоугольная сетка, а сам график помещается внутри рамки, имеющей форму прямоугольного параллелепипеда. Для отключения сетки при построении графика необходимо указать опцию *Mesh* $\rightarrow False$, для отключения обрамления – *Boxed* $\rightarrow False$. Стили линий сетки и рамки определяются соответственно опциями *MeshStyle* и *BoxStyle*. Соотношение длин сторон рамки задается опцией *BoxRatios*. На каждую грань рамки также можно нанести сетку, устанавливая опцию *FaceGrids* $\rightarrow\{face1, face2, \dots\}$. В случае *FaceGrids* $\rightarrow\{face1, face2, \dots\}$ сетка наносится только на отдельные грани рамки, причем грань определяется списком из трех чисел $\{n_1, n_2, n_3\}$, два из которых равны нулю, а третье равно 1 или -1:$

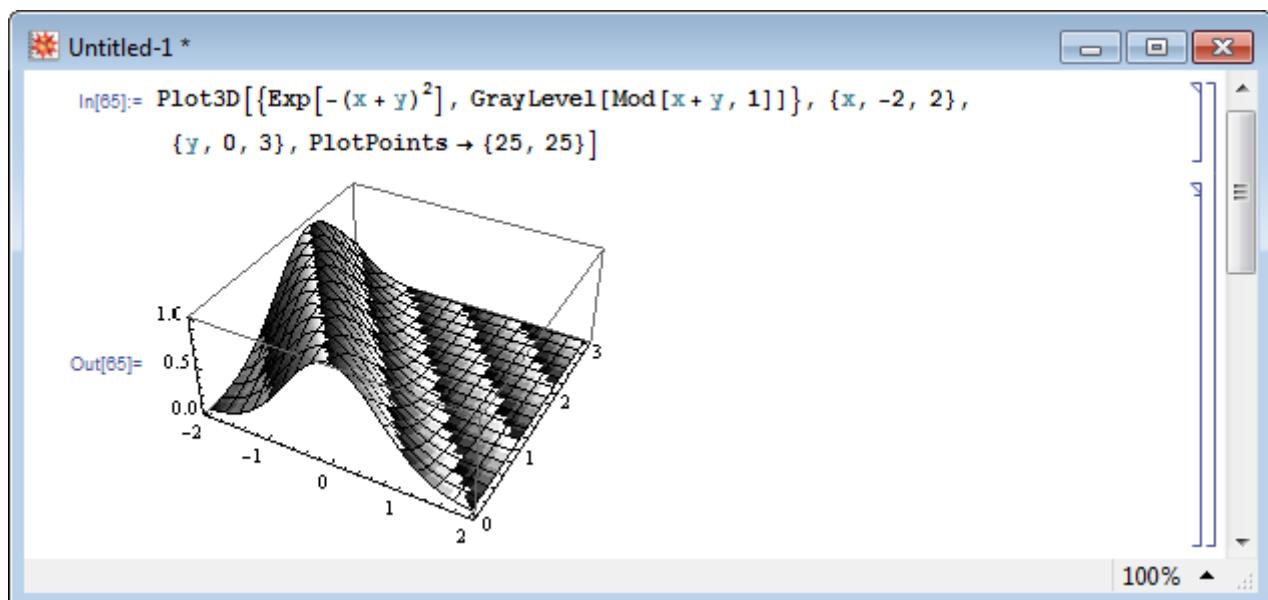


Опция *ViewPoint* определяет точку, из которой рассматривается изображаемая поверхность. По умолчанию координаты точки обзора равны $\{1.3, -2.4, 2\}$. При изменении точки обзора следует иметь в виду, что центр параллелепипеда, в котором располагается график, имеет координаты $\{0, 0, 0\}$, а относительные значения x , y и z изменяются в пределах отрезка $[-1, 1]$. Очевидно, точку обзора следует выбирать за пределами параллелепипеда, ограничивающего график:



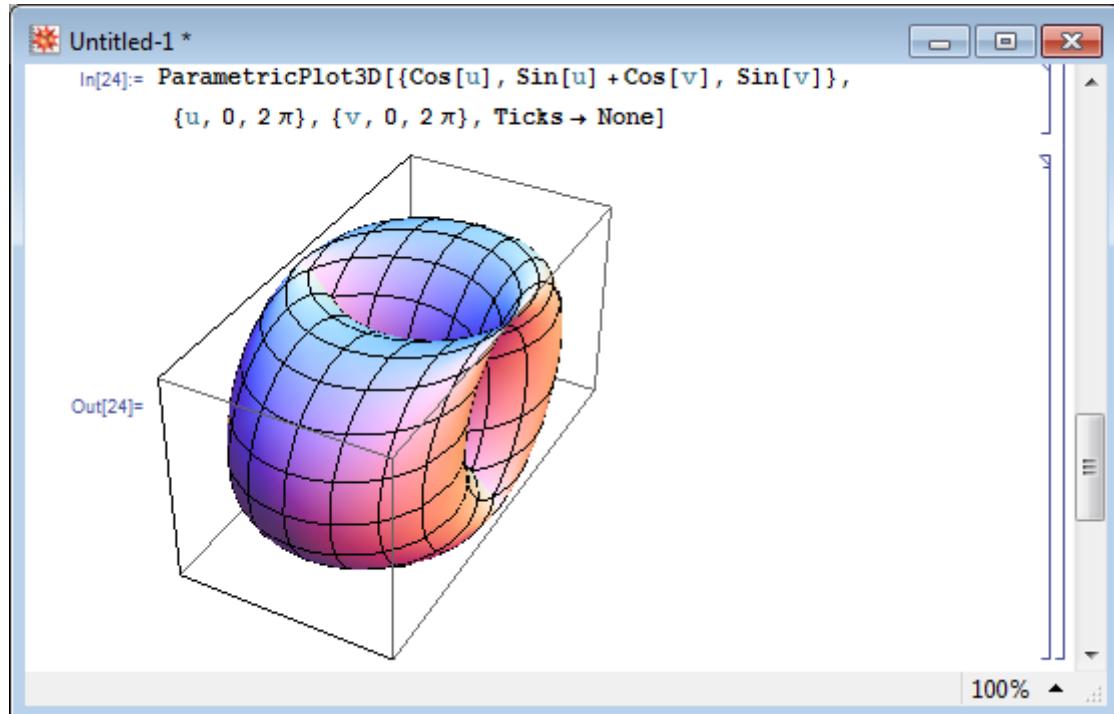
Кроме того, с помощью опции *ViewVertical* можно изменять вертикальное направление на графике (по умолчанию значение $\{0, 0, 1\}$).

По умолчанию выводимая поверхность раскрашивается, на что указывает опция *Shading* в значении *True*, причем режим раскрашивания определяется опцией *Lighting*. По умолчанию эта опция имеет значение *True*, что означает использование искусственной «подсветки». Если для опции *Lighting* установлено значение *False*, то расцветка поверхности зависит от цветовой функции, которая определяется опцией *ColorFunction*. По умолчанию в качестве световой функции используется *GrayLevel*, причем оттенок серого цвета, в который окрашивается каждая часть поверхности, зависит от значения соответствующей координаты $z=f(x,y)$. Характер расцветки поверхности можно изменить, если вместе с функцией $f(x,y)$ указать соответствующую цветовую функцию, заключив обе функции в фигурные скобки и поместив их в качестве первого аргумента функции *Plot3D*:

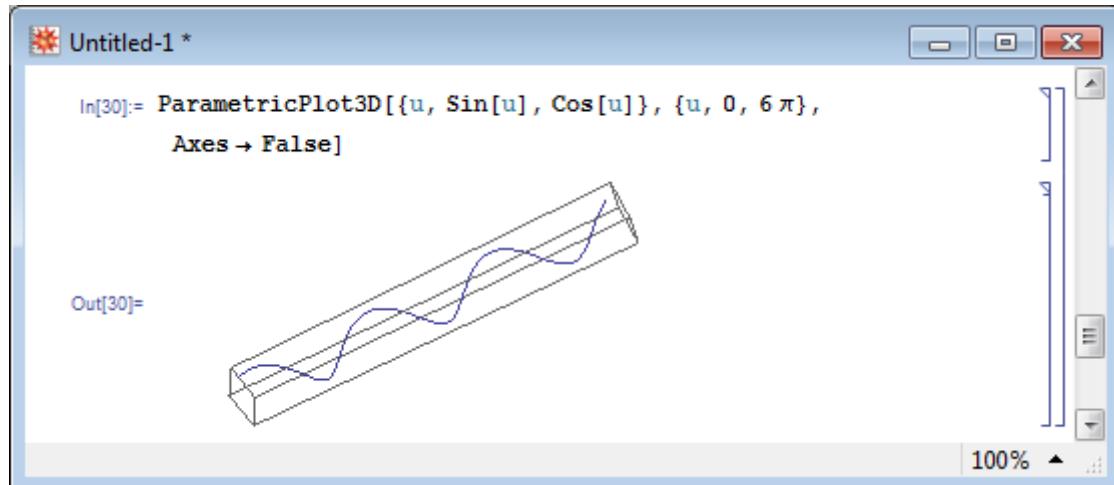


Места разрывов графика по умолчанию также раскрашиваются, чтобы отменить это действие, нужно указать опцию $\text{ClipFill} \rightarrow \text{None}$.

Если поверхность задана параметрически, то для ее визуализации используется функция ParametricPlot3D , формат обращения к которой имеет следующий вид: $\text{ParametricPlot3D}[\{fx[u,v], fy[u,v], fz[u,v]\}, \{u, umin, umax\}, \{v, vmin, vmax\}]$:

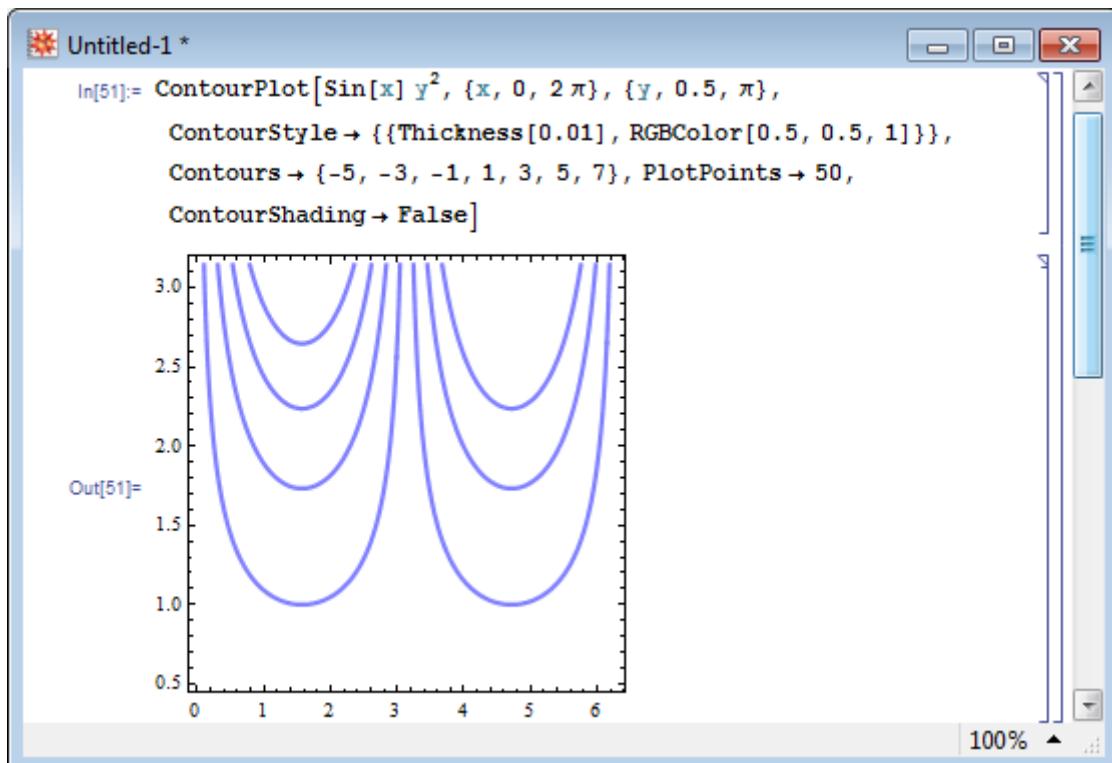


Функция ParametricPlot3D позволяет также визуализировать пространственную кривую заданную параметрически:

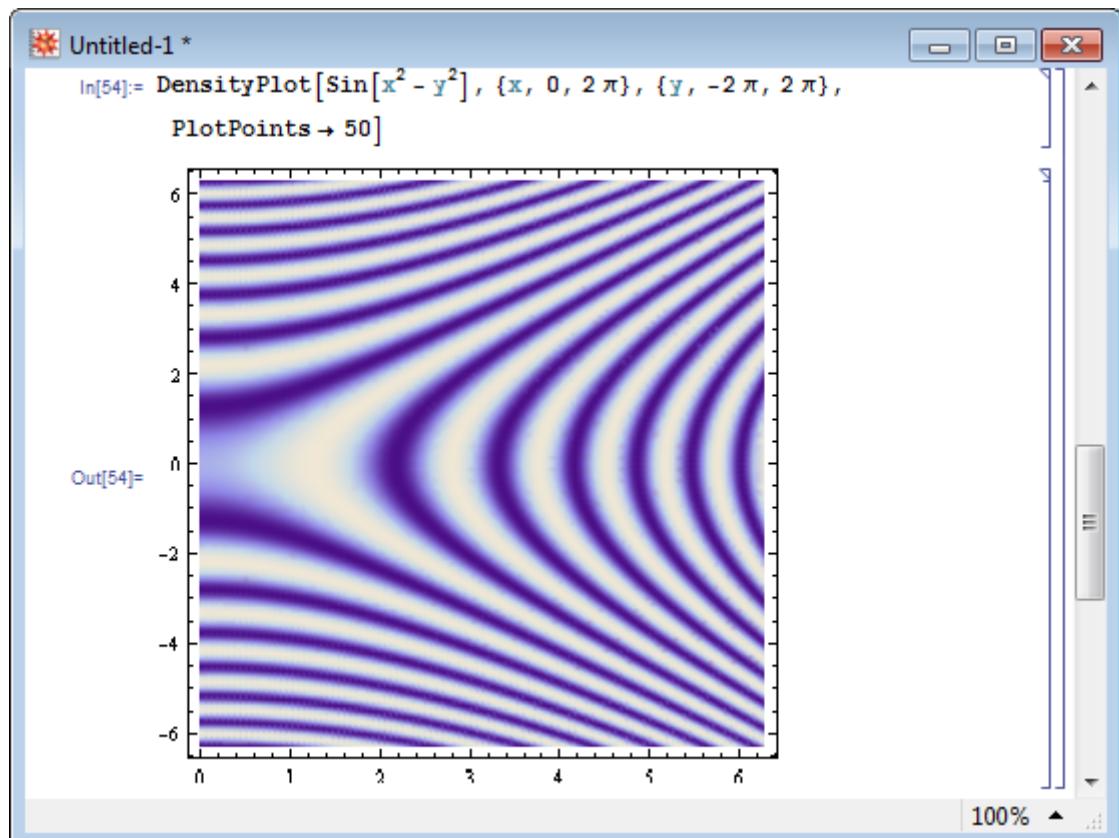


Если поверхность $z=f(x,y)$ пересечь плоскостью $z=const$, перпендикулярной к оси z , то в сечении получится плоская кривая, которую можно спроектировать на плоскость xy . Проводя множество таких плоскостей, и изображая соответствующие сечения на плоскости xy , получим изображение пространственной поверхности в виде линий уровня. Такая процедура реализуется с помощью функции ContourPlot , формат обращения к которой имеет вид: $\text{ContourPlot}[f[x,y], \{x, xmin, xmax\}, \{y, ymin, ymax\}]$. По сравнению с функциями Plot и ParametricPlot , которые также создают двумерные

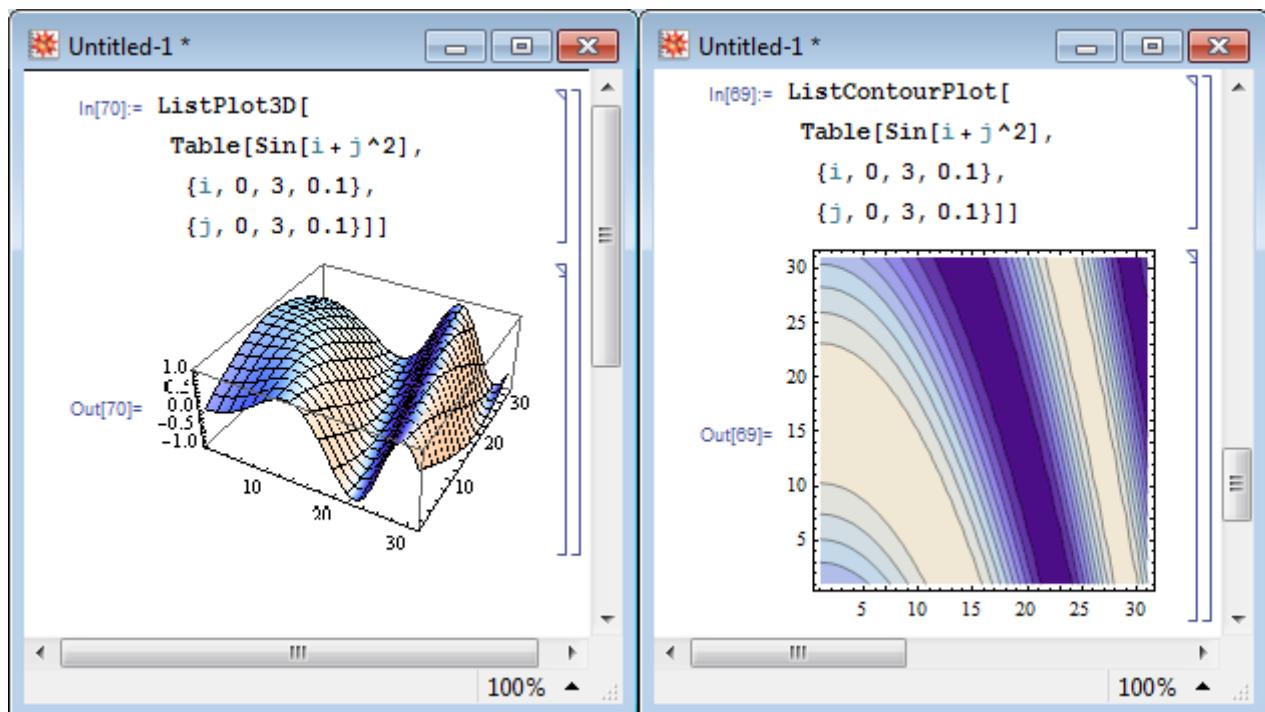
графические объекты, функция *ContourPlot* имеет несколько дополнительных опций. Так, опция *Contours* определяет количество контурных линий на графике и по умолчанию принимает значение 10. Однако для этой опции можно задать и список конкретных значений z , определяющих плоскости, пересекающие поверхность $z=f(x,y)$. Чтобы контурные линии получались достаточно гладкими, необходимо увеличивать значение опции *PlotPoints*. По умолчанию области между контурными линиями раскрашиваются различными оттенками серого цвета, причем области меньших значений функции $z=f(x,y)$ имеют более темный оттенок. Опция *ContourShading* \rightarrow *False* отменяет раскрашивание. Стиль изображения контурных линий определяется опцией *ContourStyle*:



Для двумерного представления поверхности $z=f(x,y)$ используется также функция *DensityPlot*. Область изменения каждого из аргументов разбивается на некоторое количество отрезков (которое определяется опцией *PlotPoints*), при этом плоскость xy разбивается на множество квадратных ячеек, которые раскрашиваются в соответствии с цветовой функцией, определяемой опцией *ColorFunction*. По умолчанию используется цветовая функция *GrayLevel*, причем оттенок серого цвета, в который раскрашивается ячейка, определяется значением функции $z=f(x,y)$ по правилу: ячейки, соответствующие меньшим значениям z , имеют более темный оттенок. Отметим также, что установка опции *Mesh* \rightarrow *False* приводит к исключению линий сетки на графике:



Если аналитический вид функции $z=f(x,y)$ не известен, а задано лишь множество численных значений функции в виде таблицы $\{z_{11}, z_{12}, \dots, z_{1n}\}, \{z_{21}, z_{22}, \dots, z_{2n}\}, \dots, \{z_{m1}, z_{m2}, \dots, z_{mn}\}\}$, то для визуализации можно использовать функции *ListPlot3D*, *ListContourPlot* и *ListDensityPlot*:



Задачи для самостоятельного решения

A. Построить графики функций f и g на одной координатной плоскости. Изобразить их разным цветом, типом и толщиной линии. Сделать подписи к осям координат шрифтом *Arial*, 14пт:

1. $f(x) = \sin^2 x, g(x) = x^3 - 3x + 1$
2. $f = \begin{cases} x = 3\sin 2t - \sin 3t, \\ y = 3\cos 2t + \cos 3t; \end{cases} g = \begin{cases} x = \ln(2 + t^3), \\ y = \arctant - 2t. \end{cases}$
3. $f(x) = \sqrt{x^2 + 5x - 1}, g(x) = \ln(3x + 1)$
4. $f(x) = \frac{x+1}{\sqrt{2x-3}}, g(x) = \cos \frac{x^3}{2}$
5. $f = \begin{cases} x = \arccost^2, \\ y = t \sin \sqrt{t}; \end{cases} g = \begin{cases} x = \frac{1 - \cos t}{t^2}, \\ y = \ln(1 - 2t). \end{cases}$
6. $f(x) = \tan 2x, g(x) = (3 + e^{-x})^2$
7. $f(x) = \frac{x^2}{\sqrt{4 - 5x}}, g(x) = e^x - x + 1$

B. Построить график функций f . Сделать подписи к осям координат шрифтом *Arial*, 12пт. Сделать подпись ко всему графику «График функции $f = ...$ » шрифтом *TimesNewRoman*, 14пт синего цвета:

1. $f(x, y) = \arcsin(x^2 + y^2 - 3)$
2. $f = \begin{cases} x = 5\cos t + 3\sin v, \\ y = 10\sin t \cos v - 3\cos v, \\ z = 6\sin t \sin v + 5. \end{cases}$
3. $f(x, y) = 4\sqrt{1 - x^2 - y^2}$
4. $f = \begin{cases} x = 5\cos t + 7, \\ y = 5\sin t \cos v - 3, \\ z = 5\sin t \sin v + 2. \end{cases}$
5. $f(x, y) = \frac{(x+2)^2}{2} + \frac{(y-1)^2}{6}$
6. $f = \begin{cases} x = 30 + (9 + 2\sin 5t + 2\cos v)\cos t, \\ y = 10 + (9 + 2\sin 5t + 2\cos v)\sin t, \\ z = 2\sin v. \end{cases}$
7. $f(x, y) = (x+3)^2 - \frac{(y-2)^2}{4}$

§7. НЕКОТОРЫЕ ЭЛЕМЕНТЫ ПРОГРАММИРОВАНИЯ

При проведении вычислений часто возникает необходимость условного и циклического исполнения каких-либо выражений. Для этого в пакете

Mathematica предусмотрено несколько функций. Приведем их в следующей таблице:

If[условие, действия для истины, действия для лжи]	Условный оператор
Do[тело цикла, {i, imin, imax, step}]	Простой цикл
For[i=1, условие, i++, тело цикла]	Цикл со счетчиком
While[условие, тело цикла]	Условный цикл

Если тело цикла представляет собой совокупность действий, то оно должно быть заключено в фигурные скобки «{ }»:

The image shows two side-by-side Mathematica notebooks. The left notebook (Untitled-1*) contains the following code:

```
In[137]:= a = -3; ma = 0;
If[a ≥ 0 && ma == 0,
{ma = a; a = 1;},
{ma = -a; a = -1;}];
ma
a
Out[139]= 3
Out[140]= -1

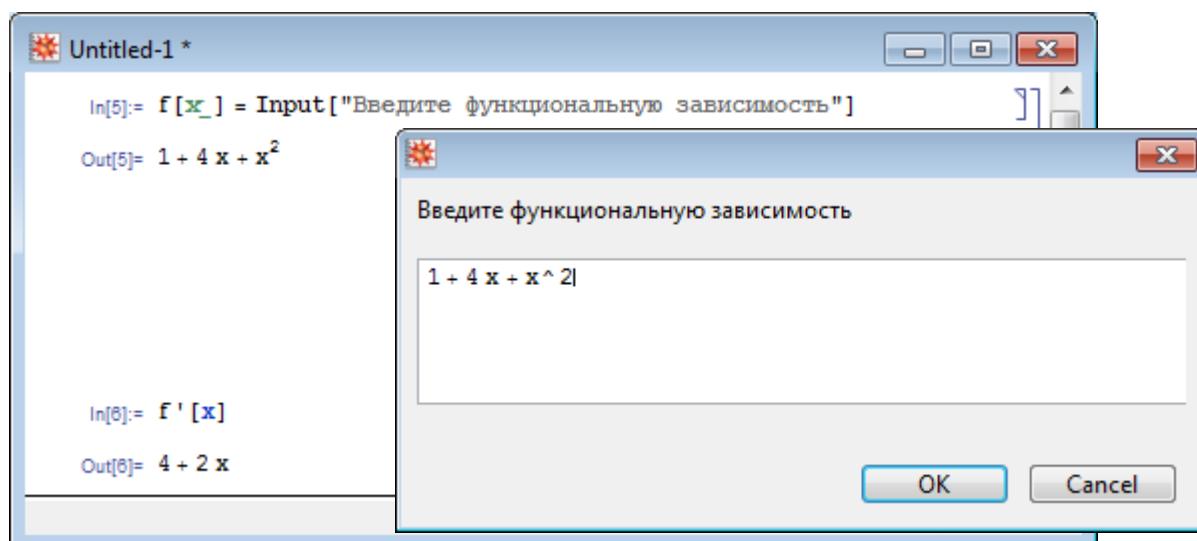
In[142]:= Do[Print[i], {i, -2, 1}]
-2
-1
0
1
```

The right notebook (Untitled-1*) contains the following code:

```
In[113]:= n = 1; p = 1;
While[n < 4, p = p n; n++];
p
Out[115]= 6

In[144]:= For[i = 1, i ≤ 5, i++,
Print[i,
" производная Sin[x]=", D[Sin[x], {x, i}]]]
1 производная Sin[x]=Cos[x]
2 производная Sin[x]=-Sin[x]
3 производная Sin[x]=-Cos[x]
4 производная Sin[x]=Sin[x]
5 производная Sin[x]=Cos[x]
```

Для ввода информации от пользователя в режиме диалога в пакете *Mathematica* встроена стандартная команда *Input[]*, которая позволяет вводить не только значения, но и выражения для функций. При инициализации команды появляется окно ввода:



ГЛАВА 2. РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Дифференциальные уравнения (ДУ) – основа математических моделей физических и других явлений и процессов. Основная цель теории ДУ – разработка методов поиска решений дифференциальных уравнений и исследование их свойств.

Уравнение называется дифференциальным, если в нем содержатся производные искомых функций или дифференциалы величин, зависимости между которыми нужно найти.

Обыкновенное ДУ имеет вид

$$F(x, y, y', \dots, y^{(m)}) = 0, \quad (1)$$

где x – независимая переменная, $y = y(x)$ – искомая функция. Порядок уравнения определяется порядком старшей производной в уравнении. В ДУ с частными производными искомая функция зависит от двух и более независимых переменных.

Решением ДУ называют такую функцию $y(x)$, которая при подстановке в уравнение обращает его в тождество. Решить обыкновенное дифференциальное уравнение – значит найти все функции, обращающие уравнение в тождество при подстановке. Для уравнения (1) семейство таких функций образуется с помощью произвольных постоянных и называется общим решением обыкновенного дифференциального уравнения n -го порядка, причем число констант совпадает с порядком уравнения: $y = y(x, C_1, C_2, \dots, C_n)$. Общее решение может быть и не разрешено явно относительно $y(x)$: $\Phi(x, y(x), C_1, C_2, \dots, C_n) = 0$. В этом случае решение принято называть общим интегралом уравнения (1). Любое конкретное решение ДУ называют его частным решением. У ДУ могут быть также особые решения, которые, как правило, не выводятся из общего решения. Графики решений ДУ называют интегральными кривыми.

Задавая некоторые фиксированные значения всем произвольным постоянным в общем решении или в общем интеграле, получаем определенную функцию, уже не содержащую произвольных констант. Эта функция называется частным решением или частным интегралом уравнения (1). Для отыскания значений произвольных постоянных, а следовательно, и частного решения, используются различные дополнительные условия к уравнению (1). Например, могут быть заданы так называемые начальные условия при $x=x_0$:

$$y(x_0) = y_0; y'(x_0) = y_0^1; \dots; y^{(n-1)}(x_0) = y_0^{(n-1)}. \quad (2)$$

В правых частях начальных условий (2) заданы числовые значения функции и производных, причем, общее число начальных условий равно числу определяемых произвольных констант. Задача отыскания частного решения уравнения (1) по начальным условиям называется задачей Коши.

Обыкновенное ДУ 1-го порядка имеет вид $F(x, y, y') = 0$, а в дифференциалах – $F(x, y, dx, dy) = 0$. Геометрически общее решение уравнения

1-го порядка представляет собой семейство кривых на плоскости XOY , отличающихся друг от друга одним параметром – значением константы C . Интегральные кривые уравнения $y' = f(x, y)$ обладают очевидным геометрическим свойством: в каждой точке (x_0, y_0) тангенс угла наклона касательной к кривой равен значению правой части уравнения в этой точке: $\operatorname{tg}\alpha = f(x_0, y_0)$. Другими словами, уравнение $y' = f(x, y)$ задает в плоскости XOY поле направлений касательных к интегральным кривым.

§1. УРАВНЕНИЯ С РАЗДЕЛЯЮЩИМИСЯ ПЕРЕМЕННЫМИ

Дифференциальным уравнением с разделяющимися переменными называется уравнение вида

$$y' = f(x)g(y) \quad (3),$$

или уравнение вида $f_1(x)g_1(y)dy + f_2(x)g_2(y)dx = 0$ в дифференциалах.

Для того, чтобы в уравнении (3) разделить переменные, то есть привести это уравнение к так называемому уравнению с разделенными переменными, необходимо произвести следующие действия:

$$\frac{dy}{dx} = f(x)g(y); \frac{dy}{g(y)} = f(x)dx, g(y) \neq 0; \int \frac{dy}{g(y)} = \int f(x)dx + C.$$

Теперь надо решить уравнение $g(y) = 0$. Если оно имеет вещественное решение $y=a$, то $y=a$ тоже будет решением уравнения (3).

Уравнение, записанное в дифференциалах, приводится к уравнению с разделенными переменными делением на произведение $f_1(x)g_2(y) \neq 0$ и решается аналогично.

Пример 1.1. Найти общее решение уравнения $y'(x) = \frac{y^2(x) - 2y(x)}{x}$.

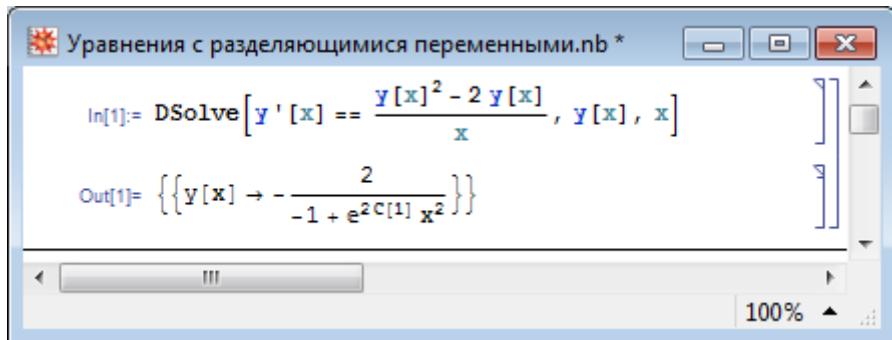
Данное уравнение можно переписать в виде $\frac{dy}{y^2(x) - 2y(x)} = \frac{dx}{x}$. Далее

интегрируем обе части уравнения и получаем

$$\begin{aligned} \int \frac{dy}{y^2 - 2y} &= \int \frac{dy}{2(y-2)} - \int \frac{dy}{2y} = \frac{1}{2} \operatorname{Log}\left(\frac{y-2}{y}\right), \int \frac{dx}{x} = \operatorname{Log}(x); \\ \operatorname{Log}\left(\frac{y-2}{y}\right) &= 2 \operatorname{Log}(x) + C. \end{aligned}$$

Следовательно, общее решение уравнения $\left(\frac{y-2}{y}\right) = Cx^2$.

Решим это уравнение в пакете *Mathematica*. Используем функцию *DSolve*



```

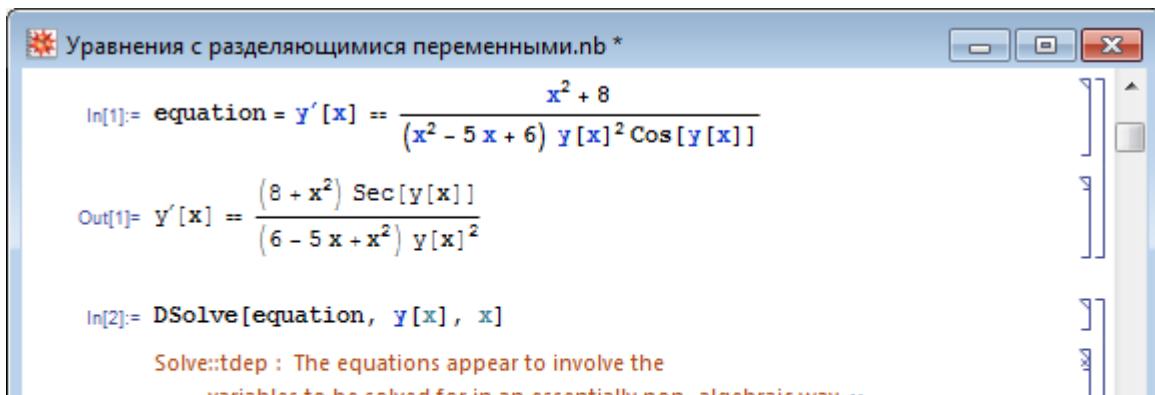
Уравнения с разделяющимися переменными.nb *
In[1]:= DSolve[y'[x] == (y[x]^2 - 2 y[x])/x, y[x], x]
Out[1]= {{y[x] \rightarrow -\frac{2}{-1 + e^{2 c[1]} x^2}}}

```

Заметим, что элементарными преобразованиями решение, полученное в пакете *Mathematica*, можно привести к решению, полученному вручную.

Пример 1.2. Решить уравнение $y' = \frac{x^2 + 8}{(x^2 - 5x + 6)y^2 \cos y}$.

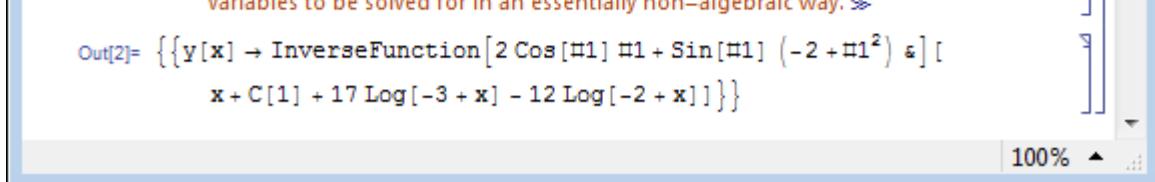
Попытаемся решить уравнение с помощью функции *DSolve*:



```

Уравнения с разделяющимися переменными.nb *
In[1]:= equation = y'[x] == (x^2 + 8)/((x^2 - 5x + 6) y[x]^2 Cos[y[x]])
Out[1]= y'[x] = (8 + x^2) Sec[y[x]] / ((6 - 5x + x^2) y[x]^2)

```

```

In[2]:= DSolve[equation, y[x], x]
Solve::tdep : The equations appear to involve the
variables to be solved for in an essentially non-algebraic way. >>
Out[2]= {y[x] \rightarrow InverseFunction[2 Cos[\#1] \#1 + Sin[\#1] (-2 + \#1^2) &][x + C[1] + 17 Log[-3 + x] - 12 Log[-2 + x]]}

```

Однако в данном случае функция *DSolve* не может решить нелинейное уравнение. Поэтому запишем уравнение в виде: $y^2 \cos y dy = \frac{x^2 + 8}{(x^2 - 5x + 6)} dx$ и будем интегрировать обе части уравнения:

```

In[3]:= (*Запишем правую и левую части*)
lhs = y^2 Cos[y];
rhs = (x^2 + 8) / (x^2 - 5 x + 6);

(*проинтегрируем*)

In[5]:= Integrate[lhs, y]
Out[5]= 2 y Cos[y] + (-2 + y2) Sin[y]

In[6]:= Integrate[rhs, x]
Out[6]= x + 17 Log[-3 + x] - 12 Log[-2 + x]

```

Следовательно, общее решение уравнения примет вид $2y\cos y + (y^2 - 2) \sin y = x + 17 \log(x-3) - 12 \log(x-2) + C$. Таким образом, мы решили уравнение в пакете *Mathematica*, сделав предварительно некоторые несложные преобразования.

Пример 1.3. Решить уравнение $\frac{dy}{dx} = \frac{x^2}{\sqrt{9 - x^2} e^y \cos y}$ с начальным условием $y(0)=0$.

Как и в предыдущем примере, сначала определим уравнение и попытаемся решить его, используя *DSolve*:

```

In[7]:= equation = y'[x] == x^2 / (Sqrt[9 - x^2] Exp[y[x]] Cos[y[x]])
Out[7]= Y'[x] =  $\frac{e^{-y[x]} x^2 \operatorname{Sec}[y[x]]}{\sqrt{9 - x^2}}$ 

In[8]:= DSolve[{equation, y[0] == 0}, y[x], x]
Solve::tdep : The equations appear to involve the
variables to be solved for in an essentially non-algebraic way. >>
DSolve::bvnul : For some branches of the general solution,
the given boundary conditions lead to an empty solution. >>
Out[8]= {}

```

В данном случае использование *DSolve* также неудачно. Разделим переменные в уравнении. Обозначим левую и правую части *lhs* и *rhs* соответственно. Проинтегрируем их отдельно:

```

In[9]:= lhs = Exp[y] Cos[y];
In[10]:= rhs = x^2 / Sqrt[9 - x^2];
In[11]:= slhs = Integrate[lhs, y]
Out[11]=  $\frac{1}{2} e^y (\cos y + \sin y)$ 
In[12]:= srhs = Integrate[rhs, x]
Out[12]=  $-\frac{1}{2} x \sqrt{9 - x^2} + \frac{9}{2} \operatorname{ArcSin}\left[\frac{x}{3}\right]$ 

```

Интерпретируя результат, заключаем, что общее решение уравнения sol имеет вид:

```

In[13]:= sol = slhs == srhs + c
Out[13]=  $\frac{1}{2} e^y (\cos y + \sin y) = c - \frac{1}{2} x \sqrt{9 - x^2} + \frac{9}{2} \operatorname{ArcSin}\left[\frac{x}{3}\right]$ 

```

Далее, чтобы решить поставленную задачу, нужно найти значение c , для которого будет выполняться начальное условие. Используем функцию *Solve* для отыскания значения произвольной постоянной, предварительно подставив в решение $y=0, x=0$:

```

In[14]:= cval = Solve[sol /. y -> 0 /. x -> 0, c]
Out[14]=  $\left\{ \left\{ c \rightarrow \frac{1}{2} \right\} \right\}$ 

```

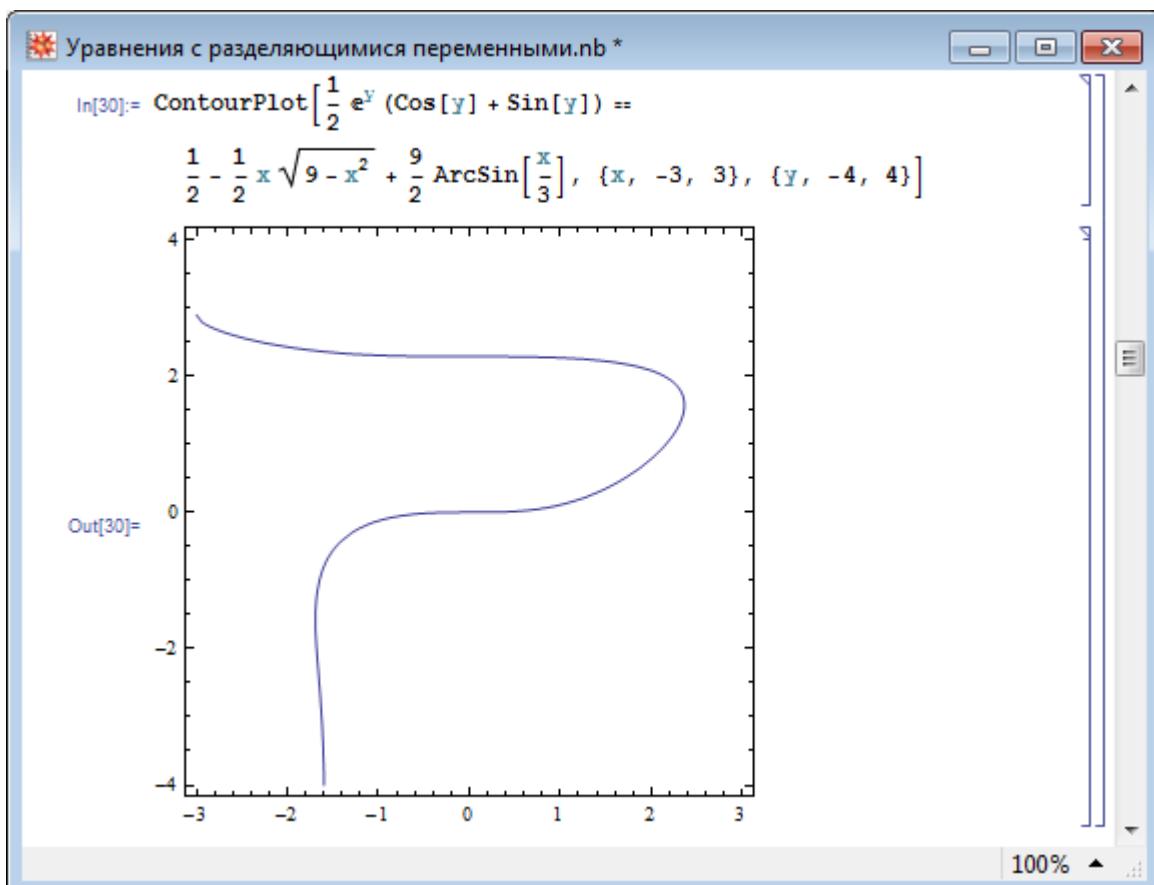
Определив значение c , получаем искомое решение в виде:

```

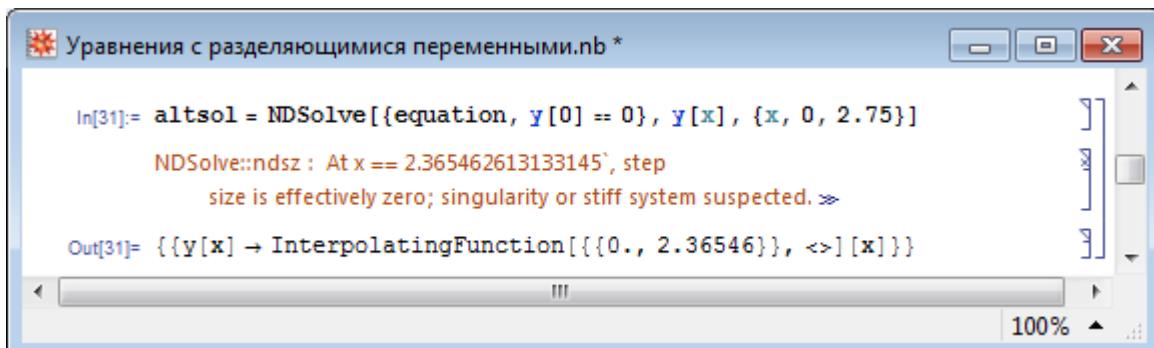
In[15]:= solution = sol /. cval[[1]]
Out[15]=  $\frac{1}{2} e^y (\cos y + \sin y) = \frac{1}{2} - \frac{1}{2} x \sqrt{9 - x^2} + \frac{9}{2} \operatorname{ArcSin}\left[\frac{x}{3}\right]$ 

```

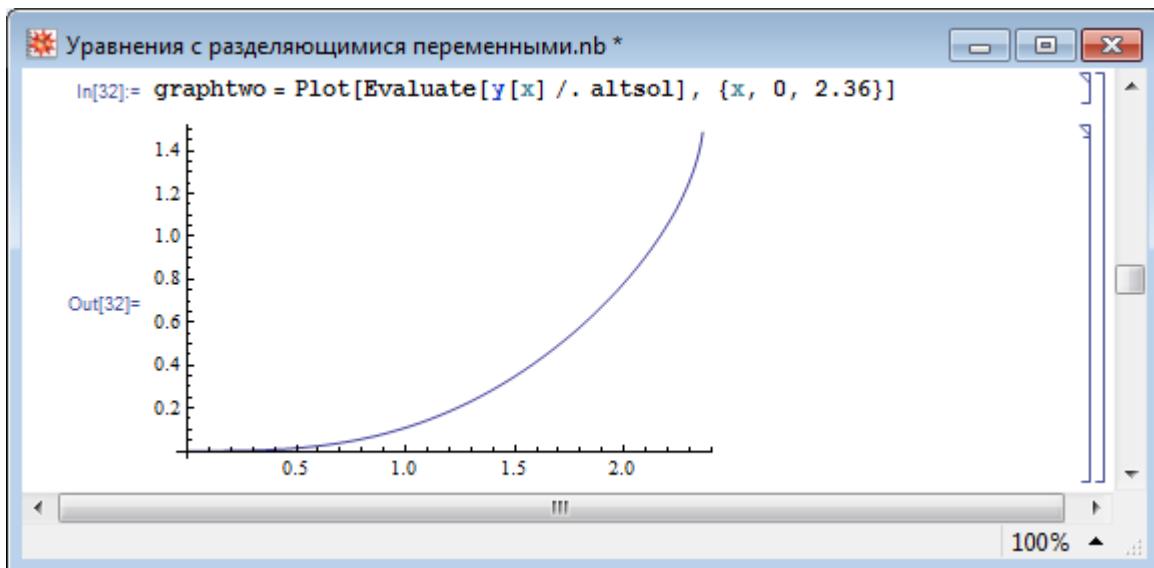
Изобразим решение графически с помощью функции *ContourPlot*:



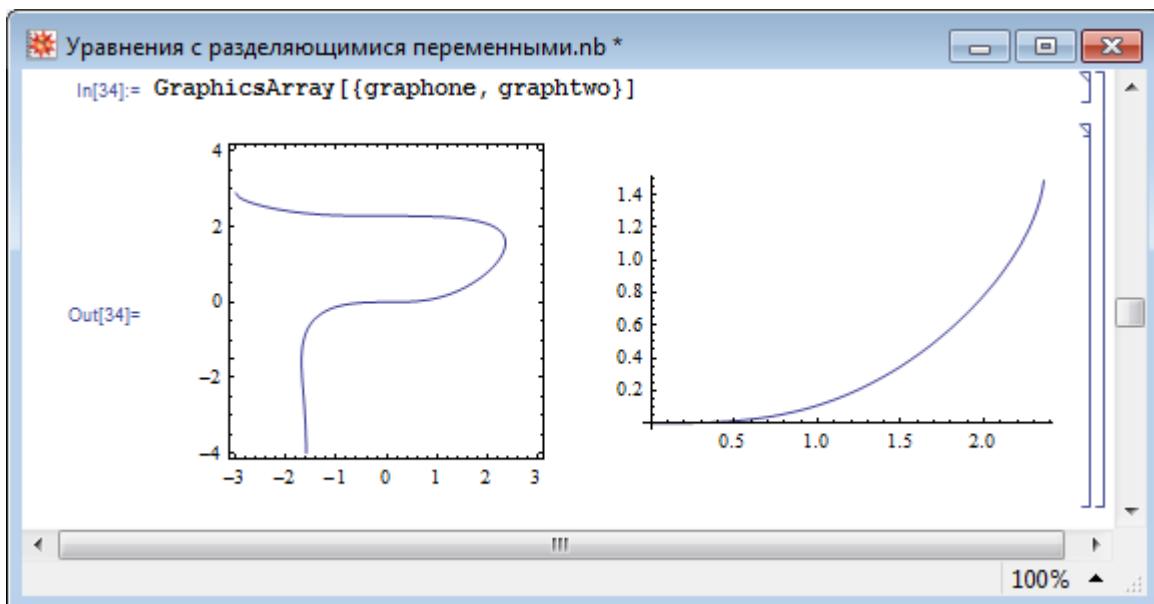
Альтернативный метод решения задачи – это использование функции *NDSolve* для численного решения задачи на данном интервале. Попытаемся использовать *NDSolve* для отыскания численного решения на интервале $0 \leq x \leq 2.75$:



Однако с помощью этой функции решение может быть найдено, но только при $x \leq 2.36$, так как дальше возникает сингулярность:



Отобразим оба решения одновременно, используя *GraphicsArray*, чтобы можно было сравнить два графика:



Задачи для самостоятельного решения

- А. Найти общие решения дифференциальных уравнений и построить несколько интегральных кривых:
1. $xydx + \sqrt{1-x^2}dy = 0$
 2. $y' = x + xy$
 3. $2x^2yy' + y^2 = 2$
 4. $y' = \frac{x^2 + 9}{(x^2 - 3x + 2)y^2 \sin y}$

5. $y' = \frac{x^2}{\sqrt{4-x^2} e^x \cos y}$
6. $y' - xy^2 = 2xy$
7. $y' = 10^{x+y}$
8. $y' = \frac{x^2 + 1}{(x^2 + 5x + 4)y^2 \cos y}$
9. $y' = \frac{x}{\sqrt{1-x^2} \sin y}$
10. $yy' + x = 1$

В. Найти частные решения дифференциальных уравнений и построить их графики:

1. $(1+y^2)dx - xydy = 0, y(1) = 0$
2. $(xy^2 + x)dy + (x^2y - y)dx = 0, y(1) = 1$
3. $(x^2 - 1)y' + 2xy^2 = 0, y(0) = 1$
4. $y' = 3\sqrt[3]{y^2}, y(2) = 0$
5. $xy' + y = y^2, y(1) = 0.5$
6. $x^2y' - \cos 2y = 1, y(+\infty) = \frac{9\pi}{4}$

§2. ОДНОРОДНЫЕ УРАВНЕНИЯ

Уравнение вида

$$M(x,y)dx + N(x,y)dy = 0 \quad (4)$$

называется однородным уравнением, если найдется такое α , что $M(tx,ty) =$

$t^\alpha M(x,y), N(tx,ty) = t^\alpha N(x,y) \quad \forall t, x, y$. Очевидно, что это уравнение всегда

может быть приведено к виду $y' = f\left(\frac{y}{x}\right)$, хотя для его решения можно этого и не делать.

Однородное уравнение приводится к уравнению с разделяющимися переменными с помощью замены искомой функции y по формуле $y=zx$, где $z(x)$ – новая искомая функция.

Замечание. Уравнение вида $y' = f\left(\frac{a_1x + b_1y + c_1}{a_2x + b_2y + c_2}\right)$ приводится к однородному уравнению некоторой заменой.

Пример 2.1. Показать, что уравнение $(x+y)dx - xdy = 0$ – однородное и решить его.

Обозначим $M(x,y) = x+y$ и $N(x,y) = -x$. Так как $M(tx,ty) = (tx) + (ty) = t(x+y) = tM(x,y)$ и $N(tx,ty) = -tx = tN(x,y)$, уравнение $(x+y)dx - xdy = 0$ – однородное уравнение первой степени.

Решим его, используя функцию *DSolve*, и получим общее решение:

```
In[5]:= DSolve[(x + y[x]) Dt[x] - x Dt[y[x]] == 0, y[x], x]
Out[5]= {{y[x] \[Rule] x C[1] + x Log[x]}}
```

Пример 2.2. Решить уравнение $(x^{\frac{1}{3}}y^{\frac{2}{3}} + x)dx + (x^{\frac{2}{3}}y^{\frac{1}{3}} + y)dy = 0$.

Обозначим $\text{capm}[x,y] = x^{\frac{1}{3}}y^{\frac{2}{3}} + x$, $\text{capn}[x,y] = x^{\frac{2}{3}}y^{\frac{1}{3}} + y$. Проверим уравнение на однородность:

```
In[1]:= capm[x_, y_] = x^(1/3) y^(2/3) + x;
In[2]:= capn[x_, y_] = x^(2/3) y^(1/3) + y;
(*проверим однородность capm*)
In[3]:= capm[t x, t y]
Out[3]= t x + (t x)^(1/3) (t y)^(2/3)
(*соберем степени*)
In[4]:= stepone = PowerExpand[capm[t x, t y]]
Out[4]= t x + t x^(1/3) y^(2/3)
```

Вынесем t в некоторой степени из выражения *stepone*, используя функцию *Collect*:

```

Однородные уравнения.nb *

steptwo = Collect[stepone, t]
t (x + x1/3 y2/3)

```

Повторим эти же действия, для *capn*:

```

(*аналогично проверим однородность capn*)

stepthree = PowerExpand[capn[t x, t y]]
t x2/3 y1/3 + t y

stepfour = Collect[steptthree, t]
t (x2/3 y1/3 + y)

```

Таким образом, исходное уравнение – однородное уравнение 1-ой степени. Применим функцию *DSolve* для решения уравнения:

```

In[7]:= sols = DSolve[capm[x, y[x]] + capn[x, y[x]] y'[x] == 0, y[x], x]
Out[7]= {y[x] → -\sqrt{-x2/3} x2/3, y[x] → (-3 x4/3 + 4 C[1])3/4 / 33/4}

```

Получим два решения. Однако мы можем решить исходное уравнение, используя стандартный алгоритм решения однородных уравнений, то есть сделать замену переменных и привести уравнение к уравнению с разделяющимися переменными. Реализуем этот алгоритм в пакете *Mathematica*.

Определим левую часть уравнения. Заметим, что функция *Dt[x]* соответствует *dx*, а *Dt[y]* – *dy*:

```

In[7]:= Clear[x, y, u]
In[8]:= leqone = capm[x, y] Dt[x] + capn[x, y] Dt[y]
Out[8]= (x + x1/3 (u x)2/3) Dt[x] + (u x + x2/3 (u x)1/3) Dt[y]

```

Сделаем замену $y=ux$, где u – новая искомая функция:

```

In[9]:= y = ux
Out[9]= ux

In[10]:= leqtwo = leqone // PowerExpand // ExpandAll
Out[10]= u1/3 x2 Dt[u] + u x2 Dt[u] + x Dt[x] + u2/3 x Dt[x] + u4/3 x Dt[x] + u2 x Dt[x]

```

Скombинируем слагаемые с $Dt[x]$, $Dt[u]$ и выделим в них отдельно множители, зависящие от x и u :

```

In[11]:= leqthree = Collect[leqtwo, {Dt[x], Dt[u], x, u}]
Out[11]= (u1/3 + u) x2 Dt[u] + (1 + u2/3 + u4/3 + u2) x Dt[x]

```

Далее мы можем решить уравнение как уравнение с разделяющимися переменными. Для этого выделим те части уравнения, на которые нужно поделить для разделения переменных:

```

In[12]:= leqthree[[1, 2]] (*первая часть уравнения, второй множитель*)
Out[12]= x2

In[13]:= leqthree[[2, 1]] (*вторая часть уравнения, второй множитель*)
Out[13]= 1 + u2/3 + u4/3 + u2

```

Разделим переменные:

```

In[14]:= leqfour = Cancel[Apart[leqthree / (leqthree[[1, 2]] leqthree[[2, 1]])]]
Out[14]= u1/3 Dt[u] / (1 + u4/3) + Dt[x] / x

```

Если приравнять это выражение к нулю, то это уравнение будет эквивалентно уравнению с разделяющимися переменными

$$\frac{\frac{1}{3}u^{1/3}Dt[u]}{1+u^{4/3}} = \frac{Dt[x]}{x}.$$

Проинтегрируем обе части полученного выражения:

```

Однородные уравнения.nb *
In[15]:= Clear[x, u, y]
In[16]:= first = Integrate[leqfour[[1, 1]] leqfour[[1, 2]], u]
Out[16]= 3/4 Log[1 + u^(4/3)]
In[17]:= second = Integrate[leqfour[[2, 1]], x]
Out[17]= Log[x]

```

Сделаем обратную замену переменных и запишем решение исходного уравнения:

```

Однородные уравнения.nb *
In[18]:= u = y/x
Out[18]= y/x
In[19]:= first == -second + constant
Out[19]= 3/4 Log[1 + (y/x)^(4/3)] == constant - Log[x]
In[20]:= solution = Exp[first] == const Exp[-second] // Simplify
Out[20]= (1 + (y/x)^(4/3))^(3/4) == const/x

```

Таким образом, мы получили общее решение исходного уравнения в виде $(1 + (\frac{y}{x})^{\frac{3}{4}})^{\frac{4}{3}} = \frac{c}{x}$, где c – произвольная константа.

Задачи для самостоятельного решения

А. Найти общие решения дифференциальных уравнений и построить несколько интегральных кривых:

$$1. \quad y' = \frac{xy + y^2 e^{-x/y}}{x^2}$$

$$2. \quad xy' - y = \sqrt{x^2 + y^2}$$

$$3. \quad 2x^3 y' = y(2x^2 - y^2)$$

$$4. \quad xy' - y = x \tan \frac{y}{x}$$

$$5. \quad (y + \sqrt{xy})dx = xdy$$

$$6. xy' = \sqrt{x^2 - y^2} + y$$

$$7. xy' = y \cos\left(\ln \frac{y}{x}\right)$$

В. Найти частные решения дифференциальных уравнений и построить их графики:

$$1. (x+2y)dx - xdy = 0, y(1) = 1$$

$$2. (y^2 - 2xy)dx + x^2 dy = 0, y(1) = 2$$

$$3. (x^2 + y^2)y' = 2xy, y(-1) = 2$$

$$4. y' = \frac{y^2 - 2xy - x^2}{y^2 + 2xy - x^2}, y(1) = -1$$

$$5. xy' - y = (x+y) \ln \frac{x+y}{x}, y(1) = 1$$

$$6. xy' = y - xe^{y/x}, y(1) = 0$$

§3. УРАВНЕНИЯ В ПОЛНЫХ ДИФФЕРЕНЦИАЛАХ

Уравнение

$$M(x,y)dx + N(x,y)dy = 0 \quad (5)$$

называется уравнением в полных дифференциалах, если его левая часть представляет собой дифференциал некоторой функции. Следовательно, если $dF(x,y) = M(x,y)dx + N(x,y)dy = 0$, то общий интеграл этого уравнения $F(x,y) = C$. Уравнение (5) является уравнением в полных дифференциалах тогда и только тогда когда $\frac{\partial M}{\partial y} \equiv \frac{\partial N}{\partial x}$.

Поскольку $\frac{\partial u}{\partial x} = M(x,y)$, то

$$u(x,y) = \int_{x_0}^x M(x,y)dx + \varphi(y), \quad (6)$$

где $\varphi(y)$ – произвольная дифференцируемая функция. Продифференцируем (6) по y :

$$\frac{\partial u}{\partial y} = \int_{x_0}^x \frac{\partial M(x, y)}{\partial y} dx + \varphi'(y). \text{ Но } \frac{\partial M}{\partial y} \equiv \frac{\partial N}{\partial x},$$

следовательно, $\frac{\partial u}{\partial y} = \int_{x_0}^x \frac{\partial N(x, y)}{\partial x} dx + \varphi'(y) = N(x, y) - N(x_0, y) + \varphi'(y)$. Положим

$$\varphi'(y) = Q(x_0, y) \Leftrightarrow \varphi(y) = \int_{y_0}^y N(x_0, y) dy, \text{ и тогда } \frac{\partial u}{\partial y} = N(x, y).$$

Итак, построена функция $u(x, y) = \int_{x_0}^x M(x, y) dx + \int_{y_0}^y N(x_0, y) dy$, для которой

$$\frac{\partial u(x, y)}{\partial x} = M(x, y), \text{ а } \frac{\partial u(x, y)}{\partial y} = N(x, y).$$

Если уравнение $M(x, y)dx + N(x, y)dy = 0$ не является уравнением в полных дифференциалах и существует такая функция $\mu = \mu(x, y)$, такая что после умножения на нее обеих частей уравнения получается уравнение $\mu(Mdx + Ndy) = 0$ в полных дифференциалах, то есть $\mu(Mdx + Ndy) = du$, то функция $\mu(x, y)$ называется интегрирующим множителем уравнения.

Если найден интегрирующий множитель μ , то интегрирование данного уравнения сводится к умножению обеих его частей на μ и к нахождению общего интеграла полученного уравнения в полных дифференциалах.

Пример 3.1. Показать, что уравнение $2xy^3dx + (1+3x^2y^2)dy = 0$ является уравнением в полных дифференциалах, а уравнение $x^2ydx + 5xy^2dy = 0$ не является таковым, и решить эти уравнения.

Рассмотрим первое уравнение. Обозначим $mm1 = 2xy^3$, $nn1 = 1+3x^2y^2$. Вычислим производную по y для $mm1$ и производную по x для $nn1$. Сравним их:

```

Уравнения в полных дифференциалах.nb *

In[1]:= 
mm1[x_, y_] = 2*x*y^3
nn1[x_, y_] = 1 + 3*x^2*y^2

Out[1]= 2*x*y^3
Out[2]= 1 + 3*x^2*y^2

In[3]:= D[mm1[x, y], y] == D[nn1[x, y], x]
Out[3]= True

```

Поскольку эти производные совпадают, то исходное уравнение является уравнением в полных дифференциалах. Попробуем решить его помощью команды *DSolve*:

```
In[5]:= DSolve[2 x y[x]^3 Dt[x] + (1 + 3 x^2 y[x]^2) Dt[y[x]] == 0, y[x], x]
Out[5]= {{y[x] -> -((2^(1/3) (9 x^4 C[1] + Sqrt[3] Sqrt[4 x^6 + 27 x^8 C[1]^2]))^(1/3) + (9 x^4 C[1] + Sqrt[3] Sqrt[4 x^6 + 27 x^8 C[1]^2])^(1/3), 2^(1/3) 3^(2/3) x^2}, {y[x] -> (1 + I Sqrt[3]) (9 x^4 C[1] + Sqrt[3] Sqrt[4 x^6 + 27 x^8 C[1]^2])^(1/3), 2^(2/3) 3^(1/3) (9 x^4 C[1] + Sqrt[3] Sqrt[4 x^6 + 27 x^8 C[1]^2})^(1/3)}, {y[x] -> (1 - I Sqrt[3]) (9 x^4 C[1] + Sqrt[3] Sqrt[4 x^6 + 27 x^8 C[1]^2])^(1/3), 2^(2/3) 3^(1/3) (9 x^4 C[1] + Sqrt[3] Sqrt[4 x^6 + 27 x^8 C[1]^2})^(1/3)}}
```

Рассмотрим второе уравнение и проведем аналогичные действия:

```
In[6]:= mm2[x_, y_] = 2 x^2 y
nn2[x_, y_] = 5 x y
Out[6]= 2 x^2 y
Out[7]= 5 x y
In[8]:= D[mm2[x, y], y] == D[nn2[x, y], x]
Out[8]= 2 x^2 == 5 y
```

Так как эти производные не совпадают, то уравнение не является уравнением в полных дифференциалах. Решим его:

```

In[9]:= DSolve[x^2 y[x] Dt[x] + 5 x y[x] Dt[y[x]] == 0,
y[x], x]

Out[9]= {y[x] → 0}, {y[x] → -x^2/10 + C[1]}

```

Пример 3.2. Решить уравнение $(2xsiny + 4e^x)dx + (x^2cosy - 2)dy = 0$ с условием $y(0)=0.5$.

Проверим, является ли уравнение уравнением в полных дифференциалах:

```

In[9]:= mm[x_, y_] = 2 x Sin[y]
nn[x_, y_] = x^2 Cos[y] - 1

Out[9]= 2 x Sin[y]

Out[10]= -1 + x^2 Cos[y]

In[11]:= D[mm[x, y], y] == D[nn[x, y], x]

Out[11]= True

```

Таким образом, исходное уравнение – уравнение в полных дифференциалах. Попробуем решить его с помощью функции *DSolve*:

```

In[12]:= DSolve[
(2 x Sin[y[x]] + 4 e^x) Dt[x] +
(x^2 Cos[y[x]] - 1) Dt[y[x]] == 0, y[x], x]

Solve::tdep :
The equations appear to involve the variables to be solved
for in an essentially non-algebraic way. >>

Out[12]= Solve[4 e^x + x^2 Sin[y[x]] - y[x] == C[1], y[x]]

```

Mathematica не может явно выразить искомую функцию и предлагает решить уравнение относительно $y(x)$. Для того, чтобы найти решение, удовлетворяющее дополнительному условию, не обязательно иметь решение в явном виде. Найдем значение произвольной постоянной из дополнительного условия:

```

In[13]:= eq = 4 ex + x2 Sin[y[x]] - y[x] == C[1]
          Solve[eq // . {y[0] → 1/2, x → 0}, C[1]]

Out[13]= 4 ex + x2 Sin[y[x]] - y[x] == C[1]

Out[14]= {C[1] → 7/2}

```

Запишем частное решение исходного уравнения, удовлетворяющее начальному условию:

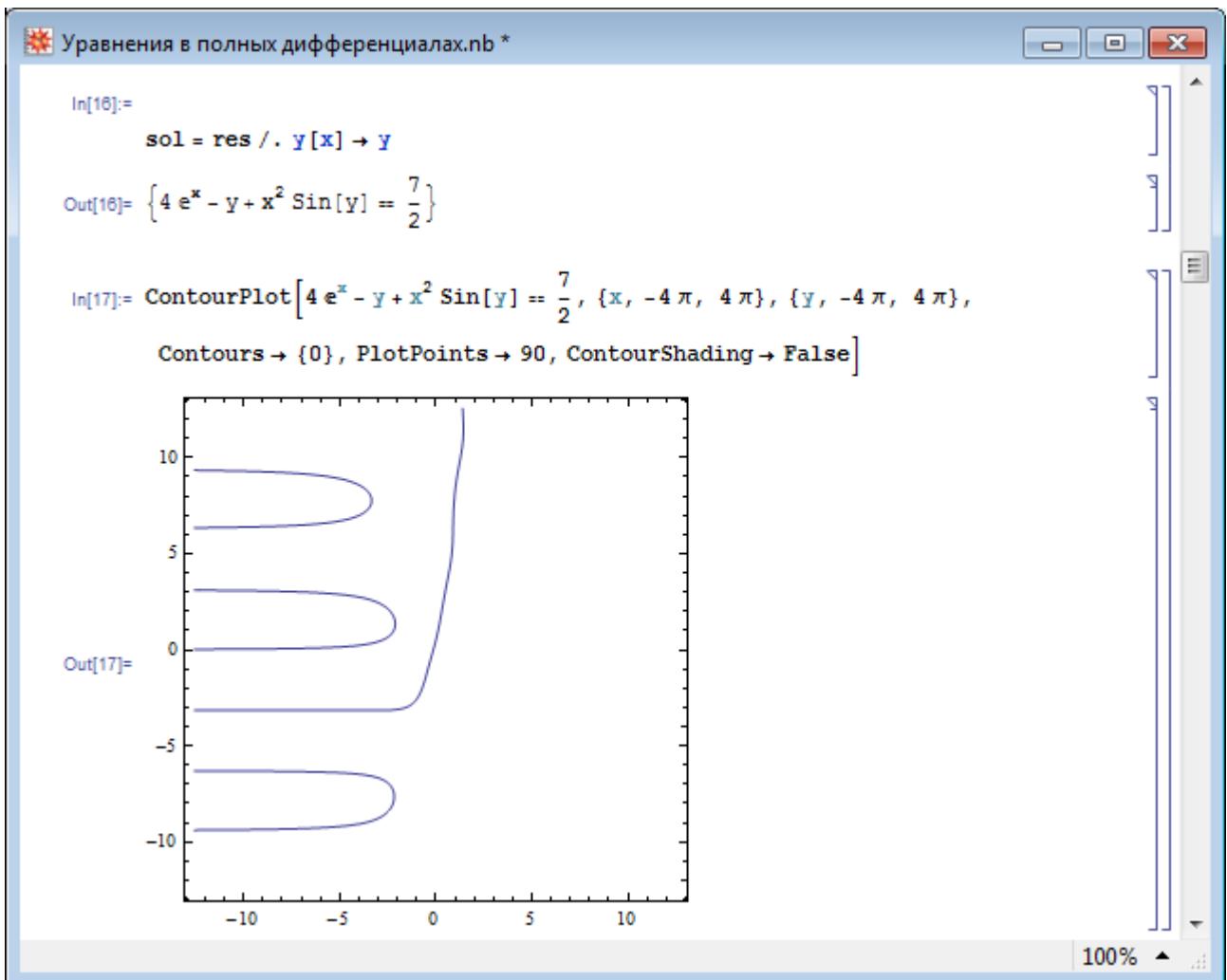
```

In[15]:= res = eq /. {#[[1]]} &

Out[15]= {4 ex + x2 Sin[y[x]] - y[x] = 7/2}

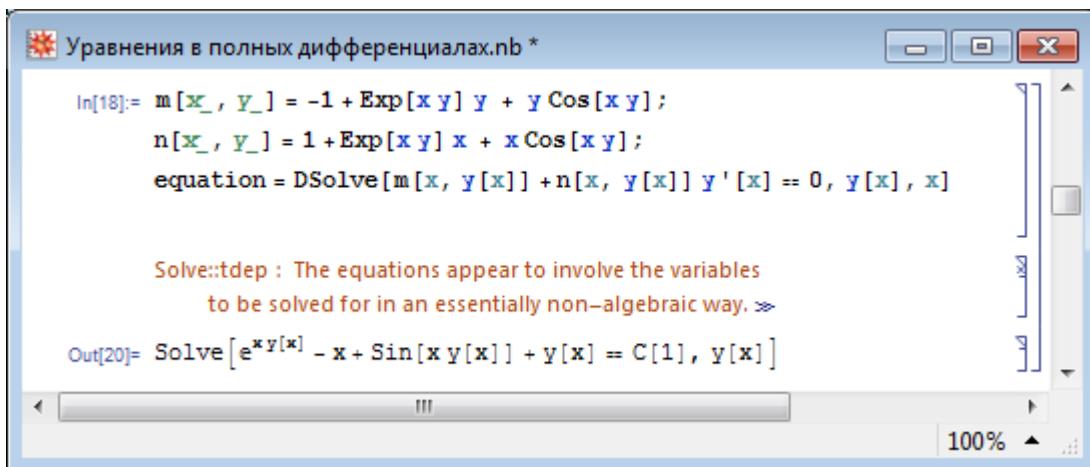
```

Построим график полученного решения в квадрате $[-4\pi, 4\pi] \times [-4\pi, 4\pi]$ с помощью функции *ContourPlot*. Укажем дополнительную опцию, указывающую только на один нулевой контур, а также опцию отмены закрашивания пространства между линиями контура *ContourShading→False*:



Пример 3.3. Найти общее решение уравнения $(-1+e^{xy}y+ycos(xy))dx + (1+e^{xy}x+ xcos(xy))dy=0$.

Определим $m(x,y) = -1+e^{xy}y+ycos(xy)$, $n(x,y) = 1+e^{xy}x+ xcos(xy)$ и попытаемся решить уравнение, используя команду *DSolve*:



В данном примере использование лишь функции *DSolve* также не приводит к желаемому результату. Поэтому попробуем решить исходное уравнение другим способом.

Вычислим производные m по y и n по x и покажем, что исходное уравнение – уравнение в полных дифференциалах:

```
In[21]:= D[m[x, y], y] == D[n[x, y], x]
Out[21]= True
```

Проинтегрируем $m(x,y)$ по x с помощью функции *Integrate* и назовем результат *stepone*:

```
In[22]:= stepone = Integrate[m[x, y], x]
Out[22]= e^x y - x + Sin[x y]
```

Полученный результат означает, что искомое решение будет иметь вид $e^{xy} - x + \sin(xy) + g(y)$. Найдем частную производную по y для полученного выражения:

```
In[23]:= steptwo = D[stepone + g[y], y]
Out[23]= e^x y x + x Cos[x y] + g'[y]
```

Так как $d(stepone + g(y)) = n(x,y)$, то используем команду *Solve*, чтобы найти значение $g'(y)$:

```
In[24]:= stepthree = Solve[steptwo == n[x, y], g'[y]]
Out[24]= {{g'[y] >= 1}}
```

Найдем $g(y)$ и общее решение уравнения:

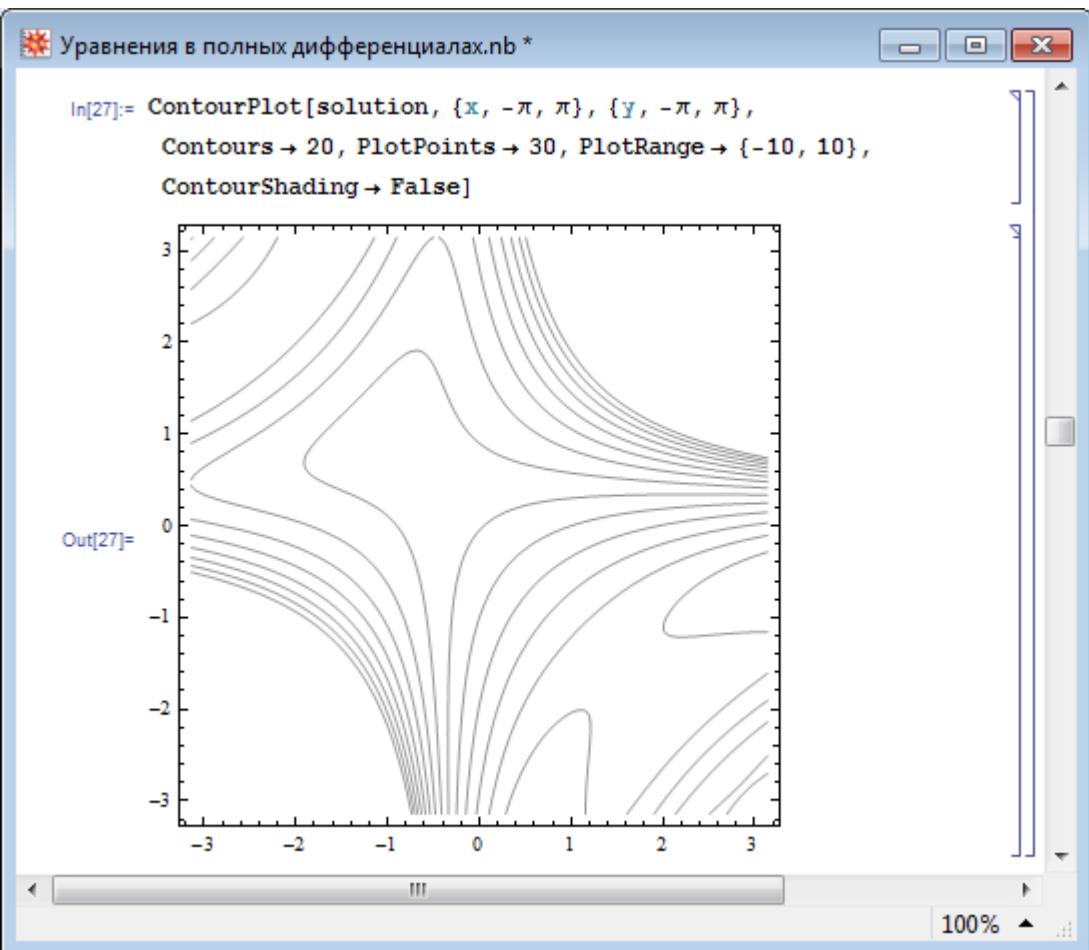
Уравнения в полных дифференциалах.nb *

```
In[25]:= stepfour = Integrate[g'[y] /. steptree[[1]], y]
Out[25]= y

In[26]:= solution = stepone + stepfour
Out[26]= exy - x + y + Sin[x y]
```

Получим, что $g(y)=y+c$ и общее решение уравнения имеет вид $e^{xy}-x+y+\sin(xy)+y=c$.

Построим интегральные кривые уравнения, то есть графики решений при различных значениях произвольной константы, с помощью функции *ContourPlot*:



Заметим, что каждая контурная линия графика соответствует некоторому значению произвольной постоянной в общем решении уравнения.

Задачи для самостоятельного решения

- А. Проверить, что данные уравнения являются уравнениями в полных дифференциалах, решить их и построить несколько интегральных кривых:

1. $2xydx + (x^2 - y^2)dy = 0$
2. $(2 - 9xy^2)xdx + (4y^2 - 6x^3)ydy = 0$
3. $e^{-y}dx - (2y + xe^{-y})dy = 0$
4. $2x(1 + \sqrt{x^2 - y})dx - \sqrt{x^2 - y}dy = 0$
5. $\left(\frac{x}{\sin y} + 2\right)dx + \frac{(x^2 + 1)\cos y}{\cos 2y - 1}dy = 0$

В. Проверить, являются ли данные уравнения уравнениями в полных дифференциалах. Если уравнения являются таковыми, решить их и найти частное решение, удовлетворяющее дополнительному условию; построить несколько интегральных кривых и найденное частное решение, отобразить их в виде массива графиков. Если уравнения не являются уравнениями в полных дифференциалах, то решить уравнения численно на интервале $[-2, 2]$ и построить график решения:

1. $\frac{y}{x}dx + (y^3 + \ln x)dy = 0, y(1) = 1$
2. $\frac{3x^2 + y^2}{y^2}dx - \frac{2x^3 + 5y}{y^3}dy = 0, y(0) = 5$
3. $y^2dx + (xy + \tan xy)dy = 0, y\left(\frac{\pi}{2}\right) = 1$
4. $(x^2 + y^2 + x)dx + ydy = 0, y(1) = 0$
5. $3x^2(1 + \ln y)dx = \left(2y - \frac{x^3}{y}\right)dy, y(1) = -1$
6. $(1 + y^2 \sin 2x)dx - 2y \cos^2 x dy = 0, y(0) = 1$
7. $xy^2(xy' + y) = 1, y(1) = 1$
8. $ydy = (xdy + ydx)\sqrt{1 + y^2}, y(1) = 0$

§4. ЛИНЕЙНЫЕ УРАВНЕНИЯ 1-ГО ПОРЯДКА

Линейным дифференциальным уравнением первого порядка называется уравнение вида $y' + p(x)y = q(x)$. Если $q(x) = 0$, то линейное уравнение называется однородным, а если $q(x) \neq 0$, то линейное уравнение называется неоднородным.

Решение однородного уравнения $y' + p(x)y = 0$ имеет вид $y = Ce^{-\int p(x)dx}$. Будем искать решение неоднородного уравнения в виде $y = C(x)e^{-\int p(x)dx}$, то есть произвольную постоянную C заменим на функцию $C(x)$. Подставим функцию $y = C(x)e^{-\int p(x)dx}$ в уравнение $y' + p(x)y = q(x)$.

Получим $C(x) = \int q(x)e^{\int p(x)dx} dx + C_1$, где C_1 – произвольная постоянная. Тогда решение

неоднородного уравнения запишется

так: $C(x) = (\int q(x)e^{\int p(x)dx} dx + C_1)e^{-\int p(x)dx}$.

Пример 4.1. Найти общее решение уравнения $x \frac{dy}{dx} + y = x \sin x$.

Поделим уравнение на x , получим $\frac{dy}{dx} + \frac{y}{x} = \sin x$ (где $p(x) = \frac{y}{x}$ и

$q(x) = \sin x$). Тогда $e^{-\int \frac{1}{x} dx} = e^{\ln|x|} = x$ для $x > 0$, и $C(x) = \int x \sin x dx + c$. Используя интегрирование по частям, где $u=x$ и $dv = \sin x dx$, получим $du = dx$, $v=-\cos x$

$C(x) = \int x \sin x dx + c = -x \cos x + \int \cos x dx + c = -x \cos x + \sin x + c$. Следовательно,

общее решение уравнения $x \frac{dy}{dx} + y = x \sin x$ для $x > 0$ имеет вид

$$y(x) = \frac{-x \cos x + \sin x + c}{x}.$$

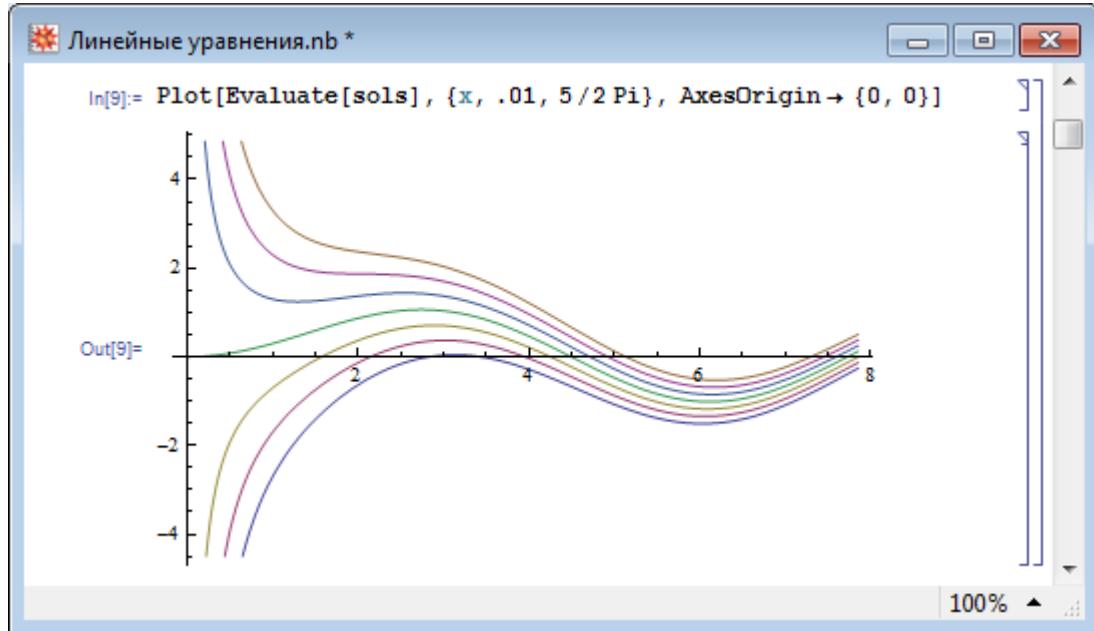
Заметим, что это же решение можно получить с помощью функции *DSolve*:

```
Линейные уравнения.nb *
In[4]:= DSolve[x y'[x] + y[x] - x Sin[x] == 0, y[x], x]
Out[4]= {{y[x] \[Rule] C[1]/x + (-x Cos[x] + Sin[x])/x}}
```

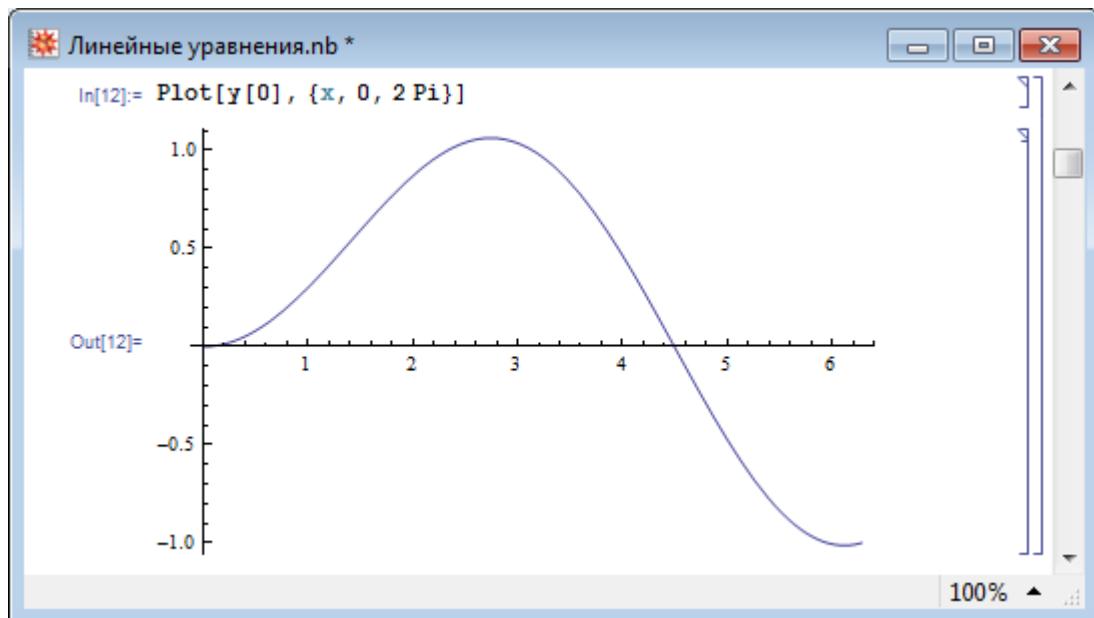
Чтобы построить график решения для разных значений произвольной постоянной c , определим y как функцию от c и создадим таблицу значений $y[c]$ для $c=-3, -2, -1, 0, 1, 2, 3$. Обозначим выходную таблицу *sols*:

```
Линейные уравнения.nb *
In[5]:= res = y[x] /. %[[1]] /. C[1] \[Rule] c;
Clear[y]
y[c_] = res
Out[7]= c/x + (-x Cos[x] + Sin[x])/x
In[8]:= sols = Table[y[c], {c, -3, 3, 1}];
```

Далее построим график $sols$ на интервале $[.01, 5\pi/2]$. Точку $x=0$ исключим, так как в этой точке решение неопределено, но опция $AxesOrigin \rightarrow \{0, 0\}$ вызвана, чтобы координатные оси пересекались в начале координат. Заметим, что решение, соответствующее значению $c=0$, не является неограниченным как другие решения вблизи точки $x=0$:



На самом деле, несмотря на то, что решение $y(0) = \frac{-x \cos x + \sin x}{x}$ неопределено в точке $x=0$, существует предел $\lim_{x \rightarrow 0} y(0) = \lim_{x \rightarrow 0} \frac{-x \cos x + \sin x + c}{x} = \lim_{x \rightarrow 0} (-\cos x + \frac{\sin x}{x}) = -1 + 1 = 0$. Поэтому *Mathematica* строит соответствующие графики правильно:



Пример 4.2. Сравнить решения уравнения $\frac{dy}{dx} + y = f(x)$ с начальным условием $y(0)=0$, при различных $f(x)=x, \sin(x), e^x$.

Mathematica может найти решение любого линейного уравнения первого порядка $y'(x) + p(x)y = q(x)$, если может вычислить интегралы: $\int p(x)dx$ и $\int q(x)e^{\int p(x)dx}dx$:

```
In[1]:= DSolve[y'[x] + p[x] y[x] == q[x], y[x], x]
Out[1]= {y[x] → e^{\int_{1-p[K[1]]}^{x-p[K[1]]} dK[1]} C[1] +
          e^{\int_{1-p[K[1]]}^{x-p[K[1]]} dK[1]} \int_1^x e^{-\int_1^{K[2]} -p[K[1]] dK[1]} q[K[2]] dK[2]}
```

В данном случае, $p(x) = 1$ и $q(x) = x, \sin(x)$ и e^x .

Чтобы вычислить решение для каждого $q(x)$, определим таблицу функций $funcs$, а потом используем команды *Table* и *Dsolve*, чтобы найти решения ДУ $y'+y=funcs[[i]]$ с начальным условием $y(0)=0$, для $i=1,2,3,4$, где $funcs[[i]]$ – i -й элемент списка функций $funcs$.

Список решений назовем *sols* и выведем в табличной форме:

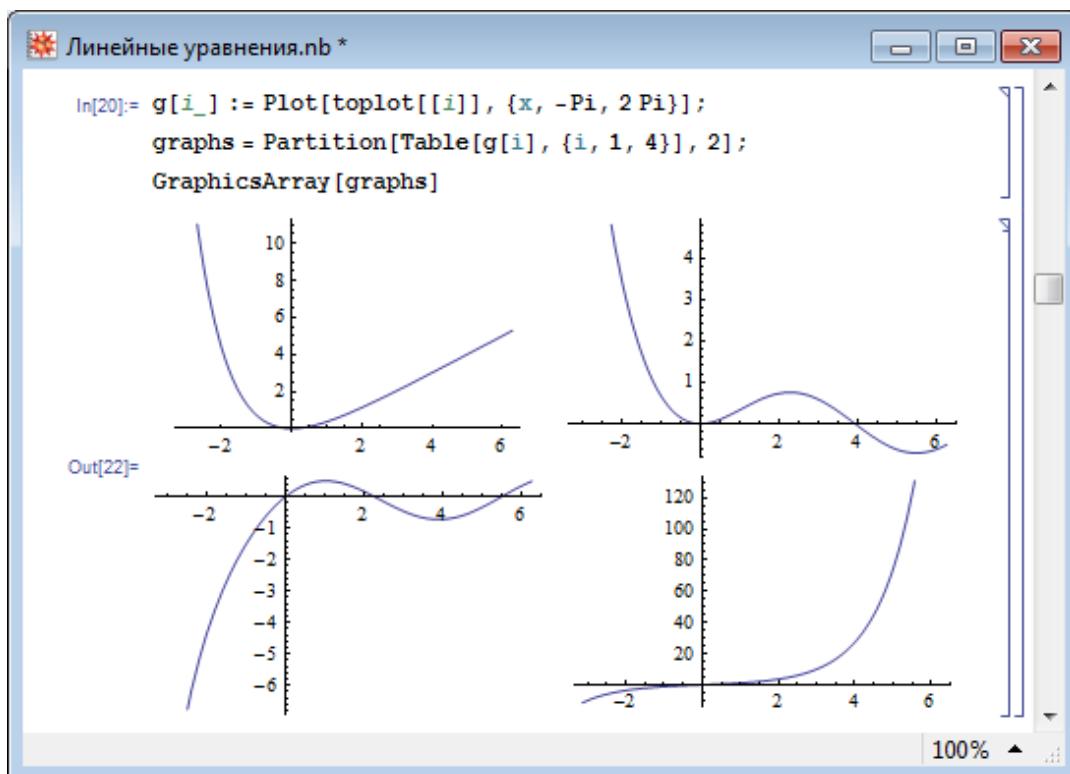
```
In[2]:= funcs = {x, Sin[x], Cos[x], Exp[x]};
sols =
Table[DSolve[{y'[x] + y[x] == funcs[[i]], y[0] == 0}, y[x], x],
{i, 1, 4}];
TableForm[sols]
Out[4]/TableForm=
y[x] → e^{-x} (1 - e^x + e^x x)
y[x] → -1/2 e^{-x} (-1 + e^x Cos[x] - e^x Sin[x])
y[x] → 1/2 e^{-x} (-1 + e^x Cos[x] + e^x Sin[x])
y[x] → 1/2 e^{-x} (-1 + e^{2x})
```

Заметим, что первый элемент *sols* – это список $\{y[x] \rightarrow -1 + e^{-x} + x\}$, который может быть получен командой *sols[[1]]*. Чтобы получить $-1 + e^{-x} + x$ в чистом виде, можно переписать выражение еще раз или извлечь из *sols*. Один из способов извлечения выражения $-1 + e^{-x} + x$ из *sols* – это запись с оператором подстановки $y[x]/.sols[[1]]$, который заменяет $y[x]$ значением $-1 + e^{-x} + x$ или ввести *sols[[1,1,1,2]]*.

Чтобы построить график каждого решения, записанного в *sols*, необходимо извлечь эти решения из *sols*. Один из способов извлечения решения – это создание таблицы значений $y[x]$, где $y[x]$ заменяется на i -й

элемент $sols$ по правилу, которое определено в следующей за знаком подстановки команде. Создадим список решений и назовем его $toplot$ для дальнейшего использования. Также каждое i -ое решение в чистом виде можно получить командой $sols[[i, 1, 1, 2]]$:

Построим график каждой функции из списка $toplot$ на интервале $[-\pi, 2\pi]$. Для удобства сравнения изобразим все четыре графика в виде графического массива с помощью функции $GraphicsArray$:



Пример 4.3. Найти общее решение уравнения $\frac{dy}{dx} - \frac{4x}{x^2 + 1}y = (1+x^2)^3 e^x$.

В данном случае, с помощью функции $DSolve$ общее решение уравнения находится корректно:

Однако можно использовать и общий метод решения линейных уравнений первого порядка. Вычислим $e^{\int \frac{-4x}{x^2+1} dx}$ и обозначим результат *intfac*:

Чтобы вычислить $\int \frac{1}{(1+x^2)^3} (1+x^2)^3 e^x dx$, воспользуемся тем, что $intfac = \frac{1}{(1+x^2)^2}$. Обозначим этот интеграл *steptwo*:

Найдем теперь общее решение уравнения и обозначим его *sol*:

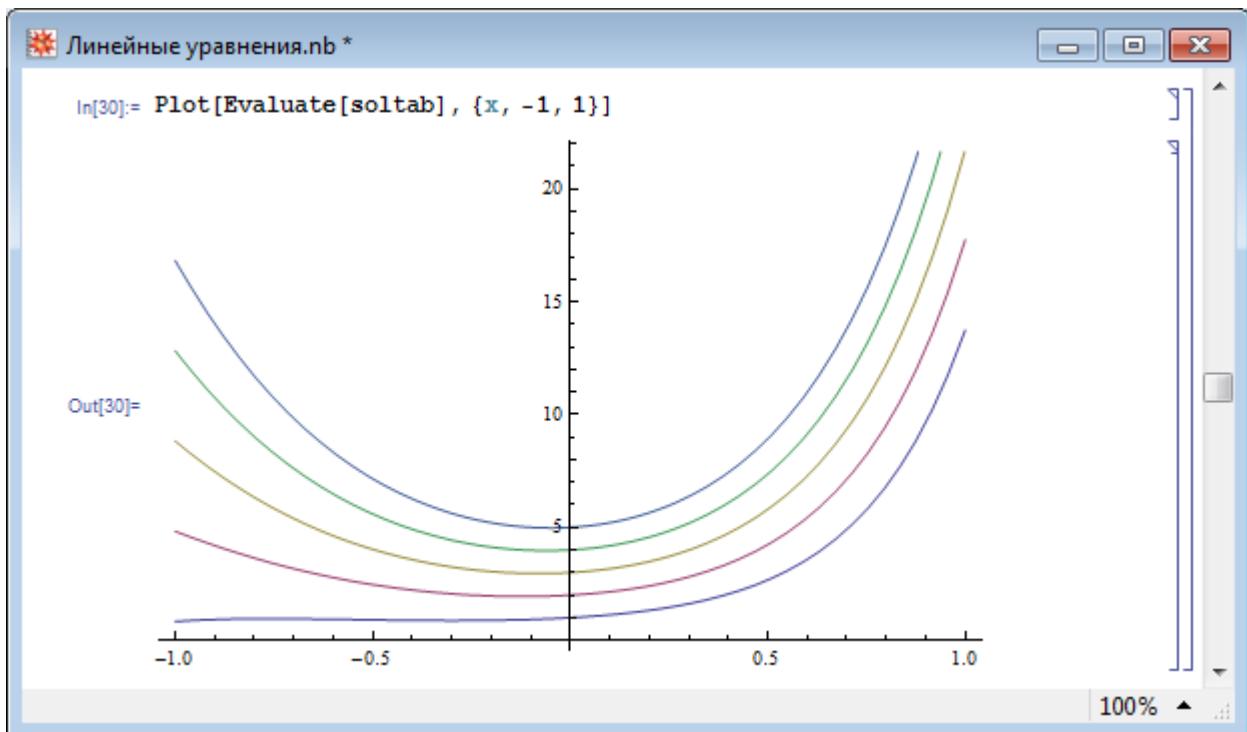
Построим графики решения для различных значений *c[i]*. Для этого создадим таблицу решений, полученных подстановкой вместо *c[i]* значений *i*=-2,-1,0,1,2 и назовем ее *soltab*:

Линейные уравнения.nb *

```
In[29]:= soltab = Table[sol /. c[i] -> i, {i, -2, 2}];
```

100%

Изобразим графически решения из таблицы $soltab$ на интервале $[-1, 1]$ с помощью команд *Plot* и *Evaluate*:



Пример 4.4. Построить график решения уравнения $\frac{dy}{dx} + \frac{x}{2 + \sin x}y = e^{-x} \cos x$ с дополнительным условием $y(0) = 0$, $-\frac{\pi}{2} \leq x \leq \frac{3\pi}{2}$.

Так как *Mathematica* не может вычислить корректно интеграл $\int \frac{x dx}{2 + \sin x}$, функция *DSolve* не может быть использована для нахождения общего решения уравнения:

Линейные уравнения.nb *

```
In[34]:= DSolve[y'[x] +  $\frac{x}{2 + \sin[x]}$  y ==  $e^{-x} \cos[x]$ , y[x], x]
DSolve::dvnoarg : The function y appears with no arguments. >>
Out[34]= DSolve[ $\frac{x y}{2 + \sin[x]} + y'[x] == e^{-x} \cos[x]$ , y[x], x]

In[35]:= Integrate[x / (2 + Sin[x]), x]
Out[35]= 
$$\frac{1}{\sqrt{3}} \left( \pi \operatorname{ArcTan}\left[\frac{1 + 2 \operatorname{Tan}\left[\frac{x}{2}\right]}{\sqrt{3}}\right] - i \left( -i (\pi - 2x) \operatorname{ArcTan}\left[\sqrt{3} \operatorname{Cot}\left[\frac{1}{4}(\pi - 2x)\right]\right] - 2i \operatorname{ArcCos}[-2] \operatorname{ArcTan}\left[\frac{\operatorname{Tan}\left[\frac{1}{4}(\pi - 2x)\right]}{\sqrt{3}}\right] + \operatorname{ArcCos}[-2] + 2 \operatorname{ArcTan}\left[\sqrt{3} \operatorname{Cot}\left[\frac{1}{4}(\pi - 2x)\right]\right] + 2 \operatorname{ArcTan}\left[\frac{\operatorname{Tan}\left[\frac{1}{4}(\pi - 2x)\right]}{\log\left[-\frac{\left(\frac{1}{2} - \frac{i}{2}\right) \sqrt{3} e^{-\frac{i x}{2}}}{2}\right]}\right] \right)$$

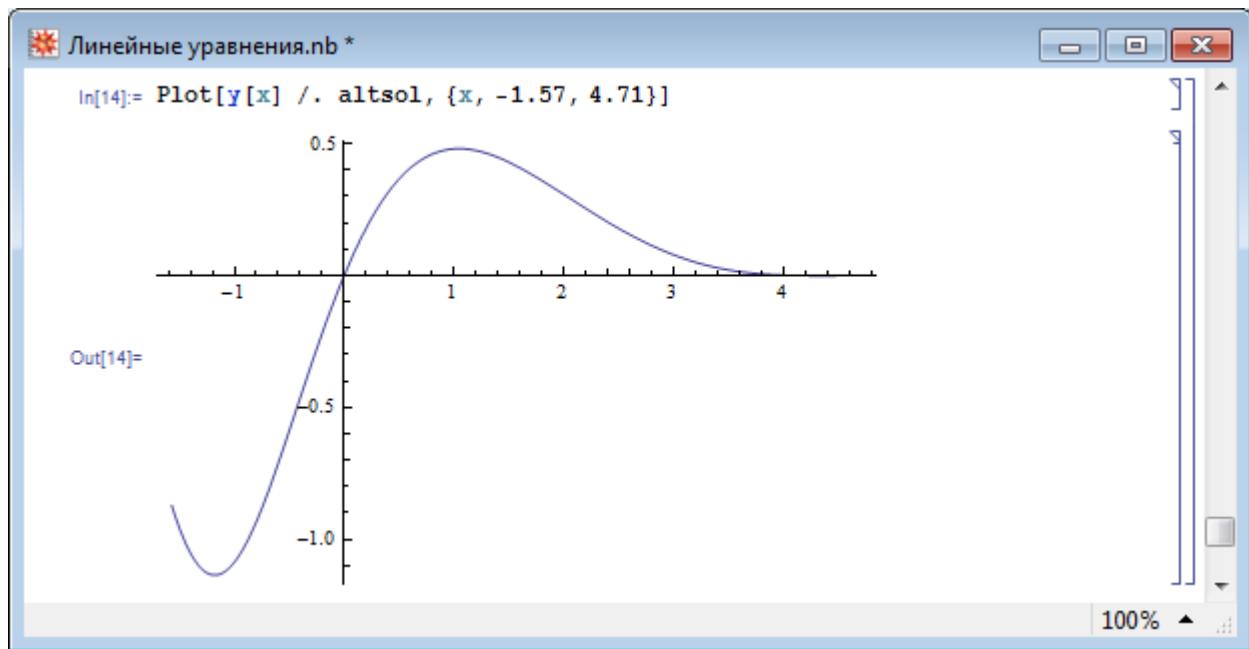
```

Альтернативным способом решения исходной задачи является численное решение дифференциальных уравнений на заданном интервале с помощью команды *NDSolve*. В этом случае *Mathematica* представляет решение в виде интерполяционной функции. Назовем ее *altsol*:

Линейные уравнения.nb *

```
In[13]:= altsol = NDSolve[{y'[x] + x y[x] / (2 + Sin[x]) == Exp[-x] Cos[x], y[0] == 0}, y[x], {x, -Pi/2, 3 Pi/2}]
Out[13]= {{y[x] \rightarrow InterpolatingFunction[{{-1.5708, 4.71239}}, < >][x]}}
```

Заметим, что в данном случае интерполяционная функция определяется на интервале $(-1.5708, 4.71239)$. Построим график решения внутри этого интервала:



Задачи для самостоятельного решения

А. Найти общие решения дифференциальных уравнений и построить несколько интегральных кривых:

1. $y' + y \tan x = \frac{1}{\cos x}$
2. $(2x+1)y' = 4x+2y$
3. $(xy + e^x)dx - xdy = 0$
4. $2x(x^2 + y)dx = dy$
5. $(x+y^2)dy = ydx$
6. $\left(\sin^2 y + \frac{x}{\tan x}\right)y' = 1$
7. $(1-2xy)y' = y(y-1)$
8. $xy' - 2y = 2x^4$

Б. Найти частные решения дифференциальных уравнений и построить их графики:

1. $x^2y' + xy + 1 = 0, y(1) = 1$
2. $y = x(y' - x \cos x), y\left(\frac{\pi}{2}\right) = \frac{\pi}{2}$
3. $(xy' - 1)\ln x = 2y, y(1) = 0$
4. $xy' + (x+1)y = 3x^2e^{-x}, y(1) = 1$
5. $xydy = (y^2 + x)dx, y(1) = 1$

$$6. \quad y' + 2y = y^2 e^x, \quad y(0) = 1$$

§5. НЕКОТОРЫЕ СПЕЦИАЛЬНЫЕ ДИФФЕРЕНЦИАЛЬНЫЕ УРАВНЕНИЯ ПЕРВОГО ПОРЯДКА

Уравнение Бернулли

Дифференциальное уравнение вида $\frac{dy}{dx} + p(x)y = q(x)y^n$, где $n \neq 0, 1$, называется уравнением Бернулли. Оно приводится к линейному уравнению с помощью замены $w(x) = (y(x))^{1-n}$. Тогда $\frac{dw}{dx} = \frac{1-n}{y^n} \frac{dy}{dx}$. Подставив в уравнение выражения для $y(x)$ и $y'(x)$ через $w(x)$, получим $\frac{y^n}{1-n} \frac{dw}{dx} + p(x)y^n w = q(x)y^n$. Умножив это уравнение на $\frac{1-n}{y^n}$, получим линейное уравнение $w' + (1-n)p(x)w = q(x)$ относительно новой искомой функции $w(x)$.

Пример 5.1. Решить уравнения:

a) $4y' - 2y = \frac{2x \sin 3x}{y}$ и b) $y' + \frac{1}{2}y = \frac{-x \sin 3x}{2}y^3$.

Оба уравнения а) и б) являются уравнениями Бернулли с $n=-1$ и $n=3$, соответственно. Замена в уравнении а) $w = y^{1-(-1)} = y^2$ приводит уравнение $4y' - 2y = \frac{2x \sin 3x}{y}$ к линейному уравнению $2w' - 2w = 2x \sin(3x)$ и, поделив на 2,

получим $w' - w = x \sin(3x)$. Вычислим интегралы $e^{\int (-1)dx} = e^{-x}$ и $\int xe^{-x} \sin 3x dx = \frac{(4-5x)\sin 3x - 3(1+5x)\cos 3x}{50e^x}$:

```

Некоторые специальные ДУ первого порядка.nb *

In[15]:= stepone = Integrate[x Exp[-x] Sin[3 x], x]
Out[15]= 1/50 e^-x (-3 (1 + 5 x) Cos[3 x] + (4 - 5 x) Sin[3 x])

```

Тогда общее решение для уравнения $w' - w = x \sin(3x)$ получим в виде $w(x) = \frac{(4-5x)\sin 3x - 3(1+5x)\cos 3x}{50} + ce^x$:

В итоге получим решение исходного уравнения, сделав обратную замену $y(x)=w(x)^{1/2}$. Обозначим его *solutionone*:

Для уравнения б) делаем подстановку $w=y^{1/3}=y^{-2}$, которая приводит уравнение $y'+\frac{1}{2}y=\frac{-x\sin 3x}{2}y^3$ к линейному уравнению $w'-w=x\sin(3x)$. Так как общее решение этого уравнения было найдено выше $w(x)=steptwo$, то для исходного уравнения б) получим решение, сделав обратную замену $y(x)=w(x)^{-1/2}$. Обозначим его *solutiontwo*:

Отметим, что эти же решения в данном случае можно получить с помощью функции *DSolve*. Но мы показали альтернативный метод решения, так как *DSolve* не для всех уравнений Бернулли работает корректно.

Чтобы построить графики решений исходных уравнений для различных значений произвольной постоянной, создадим список значений константы и таблицы решений уравнений а) и б):

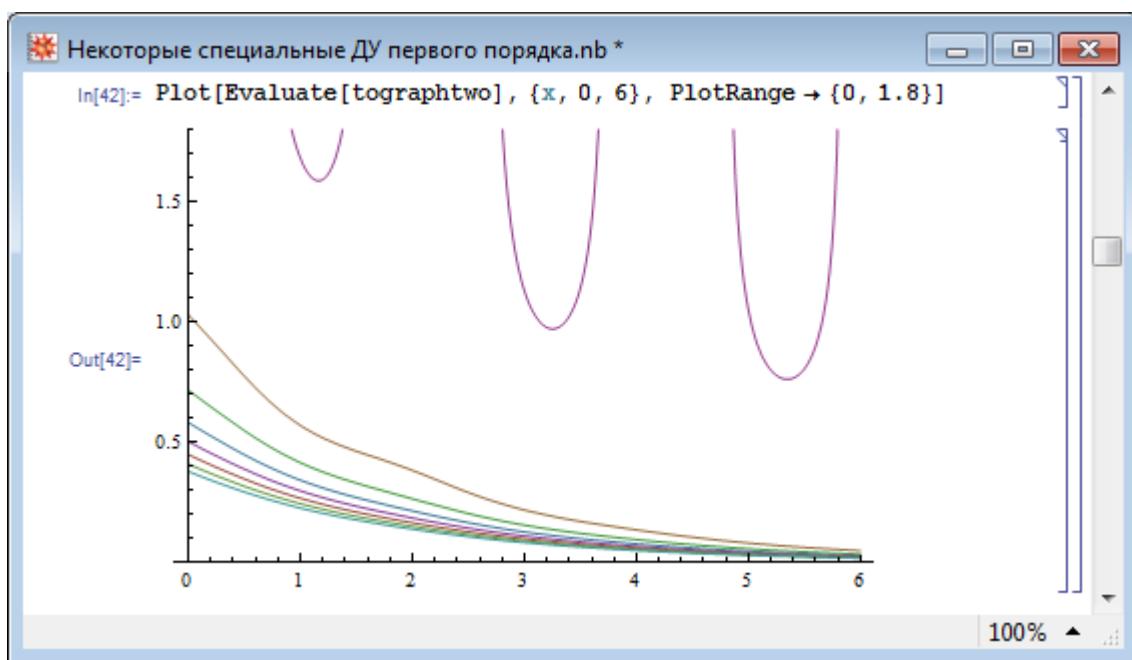
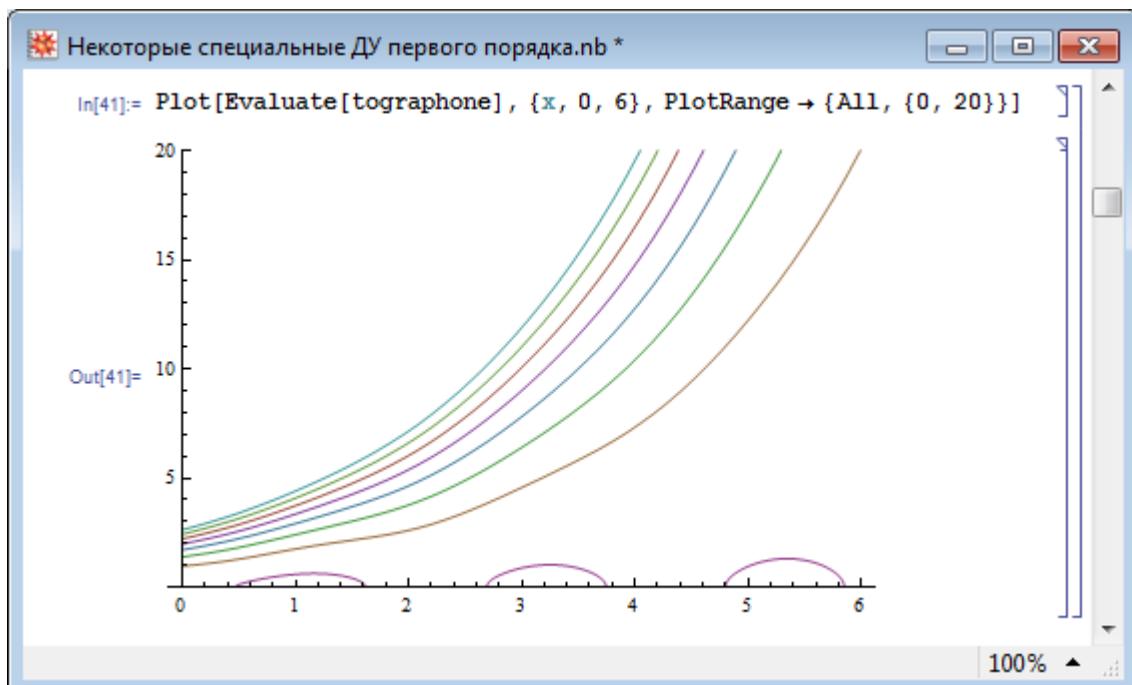
Некоторые специальные ДУ первого порядка.nb *

```
In[28]:= cs = Range[-5, 7]
Out[28]= {-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7}

In[29]:= tographone = Table[solutionone /. c → cs[[i]], {i, 1, 13}];

In[30]:= tographtwo = Table[solutiontwo /. c → cs[[i]], {i, 1, 13}];
```

Построим графики решений:



Уравнение Клеро

Уравнения вида $f(xy' - y) = g(y')$ называются уравнениями Клеро. Общее решение таких уравнений записывается в виде $f(xc - y) = g(c)$, где c – произвольная постоянная.

Пример 5.2. Решить уравнение $2(xy'(x) - y(x))^2 = (y'(x))^2 - y'(x)$.

Заметим, что это уравнение Клеро с функциями $f(x) = 2x^2 + 1$ и $g(x) = x^2 - x$. Определим функции f и g , и вычислим $f(xy'(x) - y(x))$ и $g(y'(x))$:

```
In[7]:= Clear[f, g, h]
f[x_]= 2 x^2 + 1;
g[x_]= x^2 - x;

In[10]:= lhs= f[x y'[x] - y[x]]
Out[10]= 1 + 2 (-y[x] + x y'[x])^2

In[11]:= rhs= g[y'[x]]
Out[11]= -y'[x] + y'[x]^2

In[12]:= lhs == rhs
Out[12]= 1 + 2 (-y[x] + x y'[x])^2 == -y'[x] + y'[x]^2
```

Полученный результат подтверждает, что исходное уравнение – уравнение Клеро. Функция $DSolve$ в данном случае работает не совсем корректно, а именно находит общее решение уравнения в громоздкой форме, а особое решение (которое у уравнений Клеро есть всегда) не находит вообще. Поэтому воспользуемся общим способом решения уравнений Клеро.

Как было сказано выше, общее решение в неявной форме дает формула $f(xc - y) = g(c)$, где c – произвольная постоянная. Найдем общее решение для исходного уравнения и назовем его *implicit*:

```
In[13]:= implicit= f[x c - y] == g[c]
Out[13]= 1 + 2 (c x - y)^2 == -c + c^2
```

Явное решение найдем, решив полученное уравнение относительно y с помощью функции $Solve$. Обозначим результат *explicit*:

```

In[14]:= explicit = Solve[implicit, y]
Out[14]= \{ \{ y \rightarrow \frac{1}{2} \left( -\sqrt{2} \sqrt{-1 - c + c^2} + 2 c x \right) \}, \\
           \{ y \rightarrow \frac{1}{2} \left( \sqrt{2} \sqrt{-1 - c + c^2} + 2 c x \right) \} \}

```

Корень $\sqrt{-1 - c + c^2}$ не определен, когда $-1 - c + c^2 < 0$, то есть когда c принадлежит (приближенно) интервалу $(-0.62, 1.62)$:

```

In[15]:= Solve[-1 - c + c^2 == 0 // N
Out[15]= \{ \{ c \rightarrow -0.618034 \}, \{ c \rightarrow 1.61803 \} \}

```

Явные решения можно извлечь из списка *explicit* командами *explicit[[1,1,2]]* и *explicit[[2,1,2]]*:

```

In[16]:= explicit[[1,1,2]]
          explicit[[2,1,2]]
Out[16]= \frac{1}{2} \left( -\sqrt{2} \sqrt{-1 - c + c^2} + 2 c x \right)
Out[17]= \frac{1}{2} \left( \sqrt{2} \sqrt{-1 - c + c^2} + 2 c x \right)

```

Построим графики полученного решения с различными значениями константы c . Для этого создадим список чисел $-5, -4, -3, -2, -1, 2, 4, 6, 8, 10, 12$ и 14 , используя функциями *Range* и *Union*. Заметим, что ни одно из них не лежит в интервале $(-0.62, 1.62)$:

```

In[18]:= cs = Union[Range[-5, -1], Range[2, 14, 2]]
Out[18]= \{ -5, -4, -3, -2, -1, 2, 4, 6, 8, 10, 12, 14 \}

```

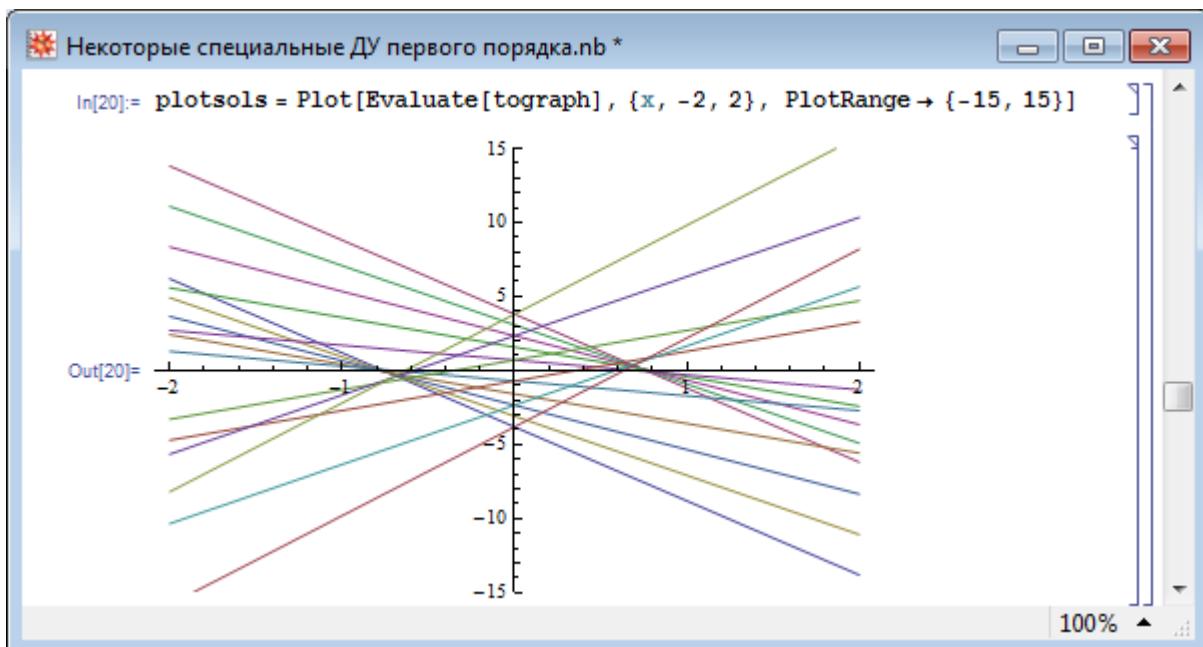
Затем создадим таблицу функций *tograph*, состоящую из явных решений, в которых константа c заменяется на значение $cs[i]$, $i=1, 2, \dots, 8$. Уберем скобки командой *Flatten*, чтобы в результате получить список функций:

Некоторые специальные ДУ первого порядка.nb *

```
In[19]:= tograph = Table[y /. explicit /. c → cs[[i]], {i, 1, 8}] // Flatten
```

```
Out[19]= {1/2 (-Sqrt[58] - 10 x), 1/2 (Sqrt[58] - 10 x), 1/2 (-Sqrt[38] - 8 x), 1/2 (Sqrt[38] - 8 x), 1/2 (-Sqrt[22] - 6 x), 1/2 (Sqrt[22] - 6 x), 1/2 (-Sqrt[10] - 4 x), 1/2 (Sqrt[10] - 4 x), 1/2 (-Sqrt[2] - 2 x), 1/2 (Sqrt[2] - 2 x), 1/2 (-Sqrt[2] + 4 x), 1/2 (Sqrt[2] + 4 x), 1/2 (-Sqrt[22] + 8 x), 1/2 (Sqrt[22] + 8 x), 1/2 (-Sqrt[58] + 12 x), 1/2 (Sqrt[58] + 12 x)}
```

Теперь построим графики этих функций на интервале $[-2, 2]$. Используем опцию *PlotRange*, чтобы определить область значений y , выводимую на экран от -15 до 15 :



Другое решение исходного уравнения Клеро, которое нельзя получить из общего решения, называется особым решением. Оно получается дифференцированием *lhs* и *rhs* по x :

Некоторые специальные ДУ первого порядка.n...

```
In[21]:= dlhs = D[lhs, x]
```

```
Out[21]= 4 x (-y[x] + x y'[x]) y''[x]
```

```
In[22]:= drhs = D[rhs, x]
```

```
Out[22]= -y''[x] + 2 y'[x] y''[x]
```

Получим уравнение $4xy''(x)(xy'(x)-y(x))=y''(x)(2y'(x)-1)$. Преобразуем его к виду $y''(x)(1-4xy(x)-2y'(x)+4x^2y'(x))=0$. Если $y''(x) \neq 0$ (отметим, что в общем решении, полученном ранее, $y''(x)=0$), то $1-4xy(x)-2y'(x)+4x^2y'(x)=0$. Левая

часть последнего уравнения извлекается из *stepone* с помощью команды *stepone[[1]]*:

```
In[23]:= stepone = Factor[dlhs - drhs]
Out[23]= (1 - 4 x y[x] - 2 y'[x] + 4 x2 y'[x]) y''[x]

In[24]:= stepone[[1]]
Out[24]= 1 - 4 x y[x] - 2 y'[x] + 4 x2 y'[x]
```

Используем функцию *DSolve* для решения уравнения $1-4xy(x)-2y'(x)+4x^2y'(x)=0$. Обозначим результат *singular*:

```
In[25]:= singular = DSolve[stepone[[1]] == 0, y[x], x]
Out[25]= {y[x] \[Rule] x/2 + Sqrt[-1 + 2 x2] C[1]}
```

Решение $\frac{x}{2} + \sqrt{-1 + 2x^2} C[1]$ получим из *singular* с помощью команды *singular[[1,1,2]]* и обозначим его *sol*:

```
In[26]:= sol = singular[[1, 1, 2]]
Out[26]= x/2 + Sqrt[-1 + 2 x2] C[1]
```

Найдем значение константы *C[1]* из условия, что *sol* должно удовлетворять уравнению $2(xy'(x)-y(x))^2=(y'(x))^2-y'(x)$. Вычислим первую и вторую производные *sol* по *x* и запишем результаты в *dsol* и *d2sol* соответственно:

Подставим в исходное уравнение $lhs=rhs$ вместо $y[x]$, $y'[x]$ и $y''[x]$ выражения sol , $dsol$ и $d2sol$ соответственно. Решим полученное уравнение $tosolve$ относительно константы $C[1]$ и обозначим список его решений $roots$:

Следовательно, особое решение исходного уравнения $y(x) = \frac{x}{2} \pm \sqrt{\frac{5}{8}} \sqrt{2x^2 - 1}$. Для дальнейшего использования получим особые решения из списка $singular$ заменой произвольных постоянных значениями, полученными в $roots$. Обозначим полученный список $singgraphs$:

Извлечем полученные решения из $singgraphs$ командами $singgraphs[[1,2]]$ и $singgraphs[[2,2]]$:

Некоторые специальные ДУ первого порядка.nb *

```
In[37]:= singgraphs[[1, 2]]
Out[37]=  $\frac{x}{2} - \frac{1}{2} \sqrt{\frac{5}{2}} \sqrt{-1 + 2x^2}$ 
```



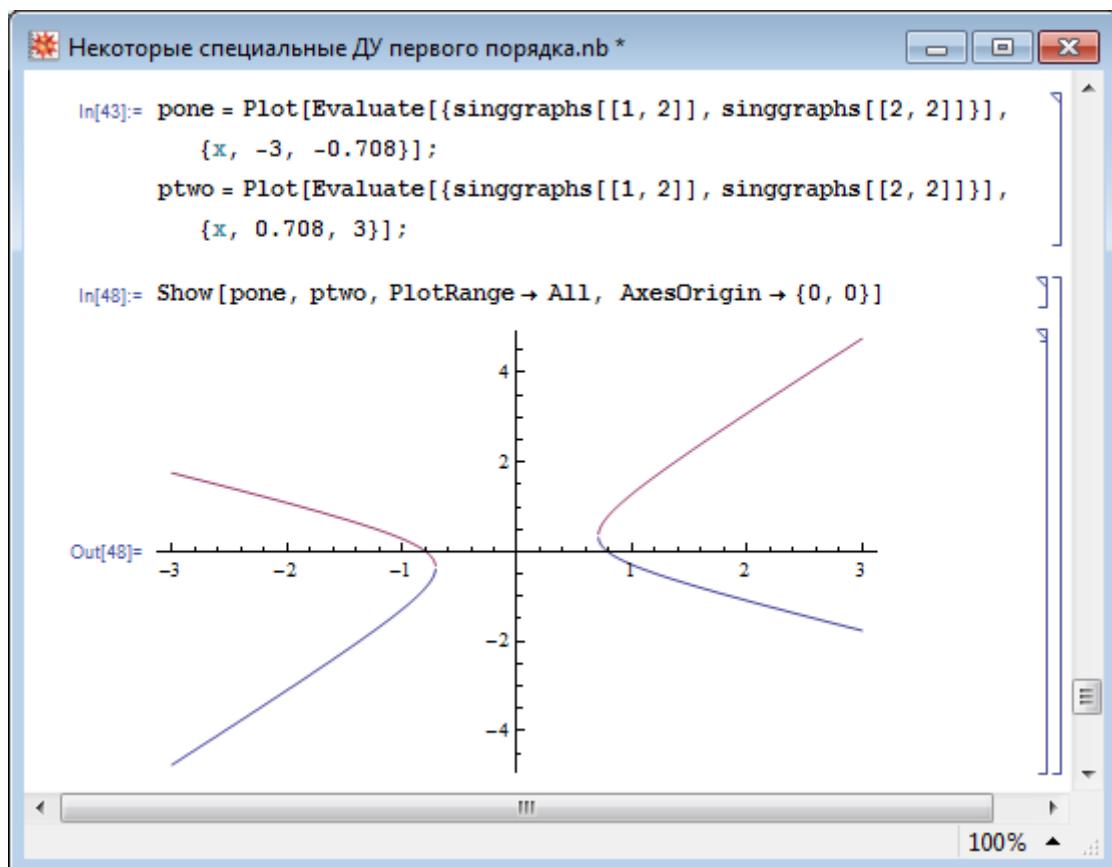
```
In[38]:= singgraphs[[2, 2]]
Out[38]=  $\frac{x}{2} + \frac{1}{2} \sqrt{\frac{5}{2}} \sqrt{-1 + 2x^2}$ 
```

Особые решения не определены, когда $2x^2 - 1 < 0$, то есть на (приближенном) интервале $(-0.708, 0.708)$:

Некоторые специальные ДУ первого порядка.nb *

```
In[39]:= Solve[-1 + 2x^2 == 0] // N
Out[39]= {{x → -0.707107}, {x → 0.707107}}
```

Нарисуем графики особых решений на интервалах $(-3, -0.708)$ и $(0.708, 3)$. Воспользуемся опцией *PlotRange* для отображения всего графика и опцией *AxesOrigin* для пересечения координатных осей в точке $(0,0)$:



Уравнение Лагранжа

Уравнения вида $y = xf(y') + g(y')$ называются уравнениями Лагранжа. Продифференцируем $y = xf(y') + g(y')$ по x . Получим $y' = xf'(y')y'' + f(y') + g'(y')y''$. Положим $p = y'(x)$ и подставим в последнее уравнение, получим $p = xf'(p) \frac{dp}{dx} + f(p) + g'(p) \frac{dp}{dx} = f(p) + \frac{dp}{dx}(xf'(p) + g'(p))$. Разрешив последнее относительно $\frac{dp}{dx}$, получим линейное уравнение $\frac{dp}{dx} = \frac{xf'(p) + g'(p)}{p - f(p)}$, которое равносильно $\frac{dp}{dx} + \frac{f'(p)}{f(p) - p}x = \frac{g'(p)}{p - f(p)}$.

Пример 5.3. Решить уравнение $y = x\left(\frac{dy}{dx}\right)^2 + 3\left(\frac{dy}{dx}\right)^2 - 2\left(\frac{dy}{dx}\right)^3$.

Заметим, что это уравнение Лагранжа с функциями $f(x) = x^2$ и $g(x) = 3x^2 - 2x^3$. Функция *DSolve* в данном случае находит решение в очень громоздкой форме, поэтому воспользуемся общим способом решения уравнений Лагранжа.

Определим функции f и g и вычислим $\frac{-f'(p)}{p - f(p)} = \frac{-2p}{p - p^2}$ и $\frac{g'(p)}{p - f(p)} = 6$,

обозначив их *stepone* и *steptthree* соответственно:

```

Некоторые специальные ДУ первого порядка.nb *

In[3]:= f[x_]:= x^2;
g[x_]:= 3 x^2 - 2 x^3;

In[5]:= stepone = -f'[p]/(p - f[p])
Out[5]= -2 p
         -----
         p - p^2

In[6]:= steptthree = g'[p]/(p - f[p]) // Cancel
Out[6]= 6

```

Далее вычислим $e^{\int stepone dp} = (p-1)^2$ и $\int steptwo stepthree dp = \int 6 e^{\int stepone dp} dp = 6p - 6p^2 + 2p^3$ и обозначим результаты *steptwo* и *stepfour* соответственно:

```

Некоторые специальные ДУ первого порядка.nb *

In[7]:= steptwo = Exp[Integrate[stepone, p]]
Out[7]= (-1 + p)^2

In[8]:= stepfour = Integrate[steptwo steptthree, p]
Out[8]= 6 p - 6 p^2 + 2 p^3

```

Запишем общее решение уравнения $\frac{dx}{dp} - x \left(\frac{f'(p)}{p-f(p)} \right) = \frac{dx}{dp} - \frac{2p}{p-p^2} x = 6$:

```

Некоторые специальные ДУ первого порядка.nb *
In[16]:= x[p_] = (stepfour + c) / steptwo
Out[16]= 
$$\frac{c + 6p - 6p^2 + 2p^3}{(-1+p)^2}$$


```

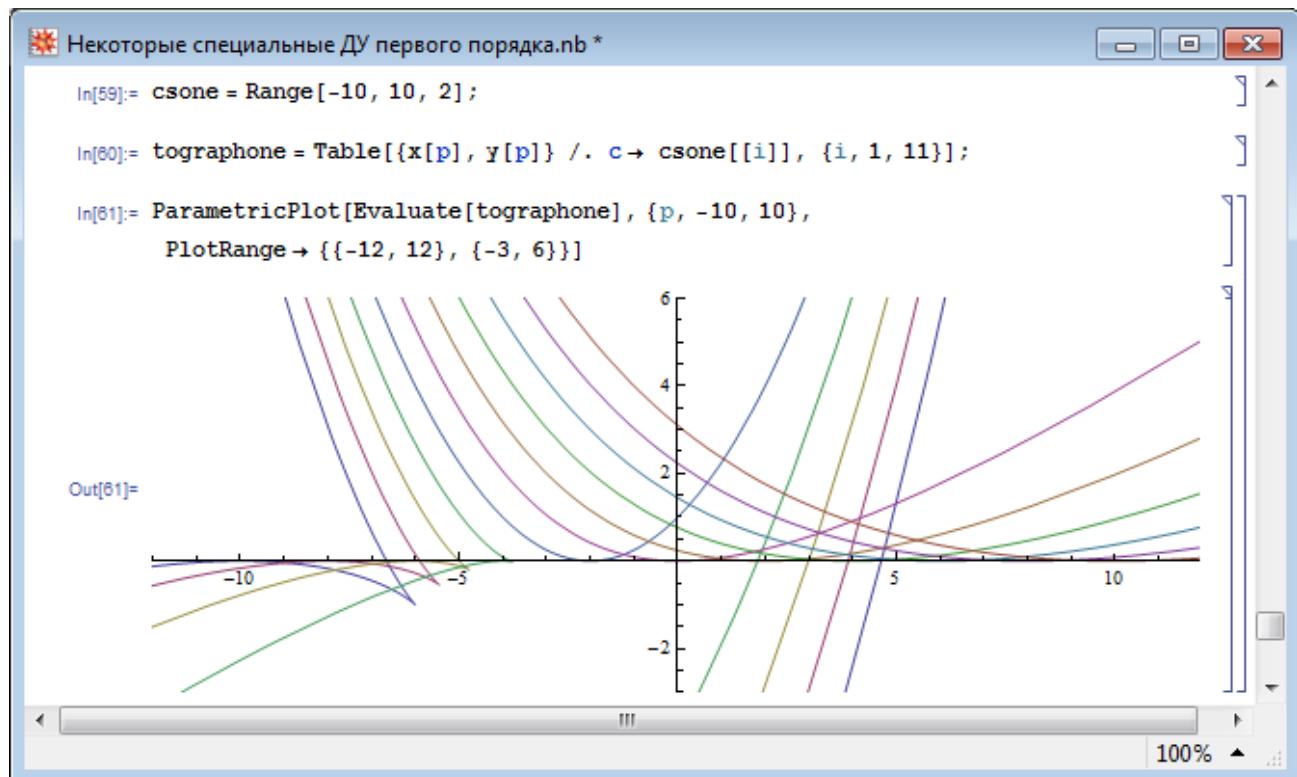
Далее запишем $y(p)$ по формуле $y(p) = x(p)f(p) + g(p)$ и получим решение исходного уравнения Лагранжа в параметрической форме $\begin{cases} x(p) \\ y(p) \end{cases}$.

```

Некоторые специальные ДУ первого порядка.nb *
In[17]:= y[p_] = x[p] f[p] + g[p]
Out[17]= 
$$3p^2 - 2p^3 + \frac{p^2(c + 6p - 6p^2 + 2p^3)}{(-1+p)^2}$$


```

Построим график решения, записанного в параметрической форме, для различных значений произвольной постоянной c . Обозначим через $csone$ набор чисел $-10, -8, \dots, 8, 10$ и определим $tographone$ как список функций $\{x(p), y(p)\}$, в котором константа c заменяется значением i -го элемента $csone$. Построим графики полученных функций с помощью команды *ParametricPlot*:



Уравнение Риккати

Уравнения Риккати – это нелинейные уравнения вида $y' + a(x)y^2 + b(x)y + c(x) = 0$.

Такое уравнение некоторой заменой приводится к уравнению второго порядка.

Положим $y(x) = \frac{w'(x)}{w(x)} - \frac{1}{a(x)}$. Тогда, $y'(x) = \frac{w''(x)}{a(x)w(x)} - \frac{(w'(x))^2}{a(x)(w(x))^2} - \frac{a'(x)w'(x)}{(a(x))^2 w(x)}$.

Подставив эти выражения в уравнение Риккати, получим уравнение второго

порядка $\frac{w''(x)}{a(x)w(x)} - \frac{a'(x)w'(x)}{(a(x))^2 w(x)} + \frac{b(x)w'(x)}{a(x)w(x)} + c(x) = 0$. Умножив уравнение на

$a(x)w(x)$ и упростив, получим уравнение $w'' - \left(\frac{a'(x)}{a(x)} - b(x) \right) w' + a(x)c(x)w = 0$.

Пример 5.4. Преобразовать уравнение Риккати

$$y' + (x^4 + x^2 + 1)y^2 - \frac{2(1 - x + x^2 - 2x^3 + x^4)}{1 + x^2 + x^4}y + \frac{1}{x^4 + x^2 + 1} = 0$$

к дифференциальному уравнению 2-го порядка. Решить уравнение Риккати с начальным условием $y(0) = 0$ на интервале $[-0.5, 1]$ численно и построить график решения.

Начнем с определения $a(x)$, $b(x)$ и $c(x)$:

```

Некоторые специальные ДУ первого порядка.nb *

In[12]:= a[x_] = x^4 + x^2 + 1;
           b[x_] = -2 (1 - x + x^2 - 2 x^3 + x^4) / (1 + x^2 + x^4);
           c[x_] = 1 / (x^4 + x^2 + 1);

```

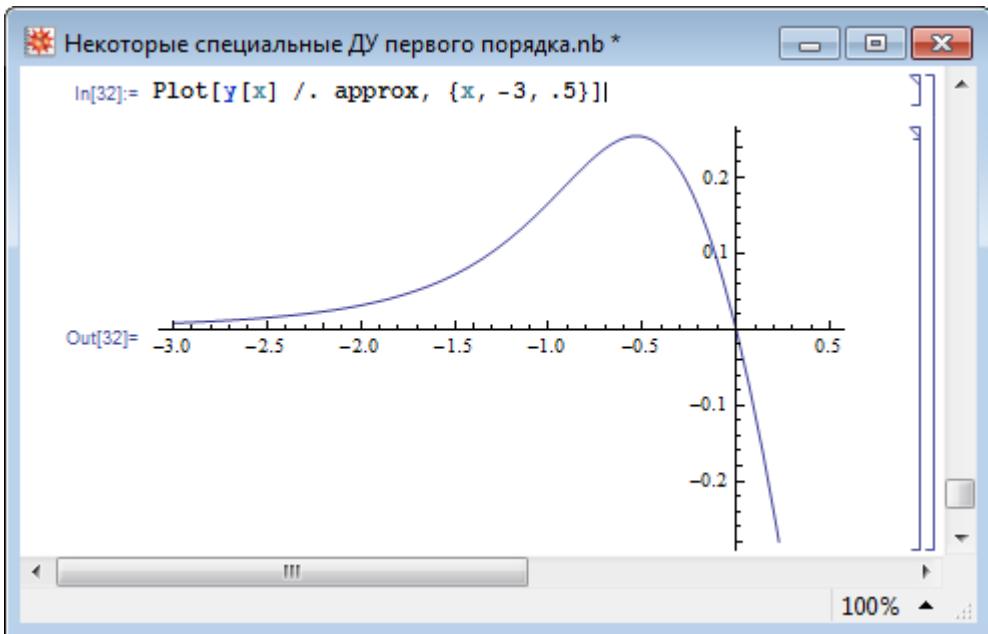
Замена $y(x) = \frac{w'(x)}{w(x)} - \frac{1}{a(x)}$ приводит к уравнению 2-го порядка:

Умножим последнее уравнение на $w(x)a(x)$ и упростим его:

Таким образом, исходное уравнение Риккати сведено к уравнению второго порядка вида $w''(x) - 2w'(x) + w(x) = 0$ относительно новой искомой функции $w(x)$.

Найдем приближенное решение данного уравнения Риккати с начальным условием $y(0)=0$ на интервале $[-0.5, 1]$ с помощью функции *NDSolve*:

Построим график полученного решения:



Задачи для самостоятельного решения

А. Найти общие решения уравнений Бернулли и построить несколько интегральных кривых. Для уравнений с дополнительным условием найти частное решение и построить его график. Оба графика отобразить в виде массива графиков:

$$1. \quad y' - \frac{y}{x-1} = \frac{y^3}{x-1}$$

$$2. \quad 3y' = (1-3y^3)y \sin x, \quad y\left(\frac{\pi}{2}\right) = 1$$

$$3. \quad (x+1)(y'+y^2) = -y$$

$$4. \quad 2y' - \frac{x}{y} = \frac{xy}{x^2-1}, \quad y(2) = 0$$

$$5. \quad xy^2y' = x^2 + y^3$$

$$6. \quad xy' - 2x^2\sqrt{y} = 4y, \quad y(1) = 0$$

В. Найти общие и особые решения уравнений Клеро и Лагранжа, построить их графики. Отобразить графики на одной координатной плоскости:

$$1. \quad y = xy' - y^2$$

$$2. \quad y = 2xy' - 4y^3$$

$$3. \quad y'^3 = 3(xy' - y)$$

$$4. \quad xy' - y = \ln y'$$

$$5. \quad 2y'^2(y - xy') = 1$$

6. $y + xy' = 4\sqrt{y'}$
7. $y = xy' - (2 + y')$
8. $y = xy'^2 - 2y'^3$
9. $xy'(y'+2) = y$
10. $2xy' - y = \ln y'$

C. Привести уравнения Риккати к уравнениям второго порядка. Решить уравнения численно на интервале $[-2, 2]$ и построить графики решений:

1. $x^2 y' + xy + x^2 y^2 = 4, y(1) = 0$
2. $3y' + y^2 + \frac{2}{x^2} = 0, y(1) = 0$
3. $xy' - (2x+1)y + y^2 = -x^2, y(1) = 0$
4. $y' - 2xy + y^2 = 5 - x^2, y(0) = 0$
5. $y' + 2ye^x - y^2 = e^{2x} + e^x, y(0) = -1$

ГЛАВА 3. ПРИЛОЖЕНИЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ 1-ГО ПОРЯДКА

§1. ОРТОГОНАЛЬНЫЕ ТРАЕКТОРИИ

Две прямые L_1 и L_2 с угловыми коэффициентами m_1 и m_2 соответственно перпендикулярны, если их угловые коэффициенты удовлетворяют соотношению $m_1 = -\frac{1}{m_2}$. Две кривые C_1 и C_2 ортогональны в точке, если соответствующие касательные к кривым перпендикулярны в этой точке.

Пример 1.1. Используя определение ортогональности, проверить, ортогональны ли кривые $y=x$ и $y=\sqrt{1-x^2}$ в точке $\left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$.

Угловой коэффициент касательной к кривой в некоторой точке определяется значением производной в данной точке. Найдем производные данных функций и обозначим их $derivy1$ и $derivy2$ соответственно:

```
Приложения ДУ 1-го порядка.nb *
In[1]:= y1[x_] := x;
          y2[x_] := Sqrt[1 - x^2];
In[5]:= derivy1 = y1'[x]
Out[5]= 1
In[6]:= derivy2 = y2'[x]
Out[6]= -x/Sqrt[1 - x^2]
```

Подставим значение $x = \frac{\sqrt{2}}{2}$ в $derivy2$ и проверим соотношение между угловыми коэффициентами:

```

In[8]:= m1 = derivy1
Out[8]= 1

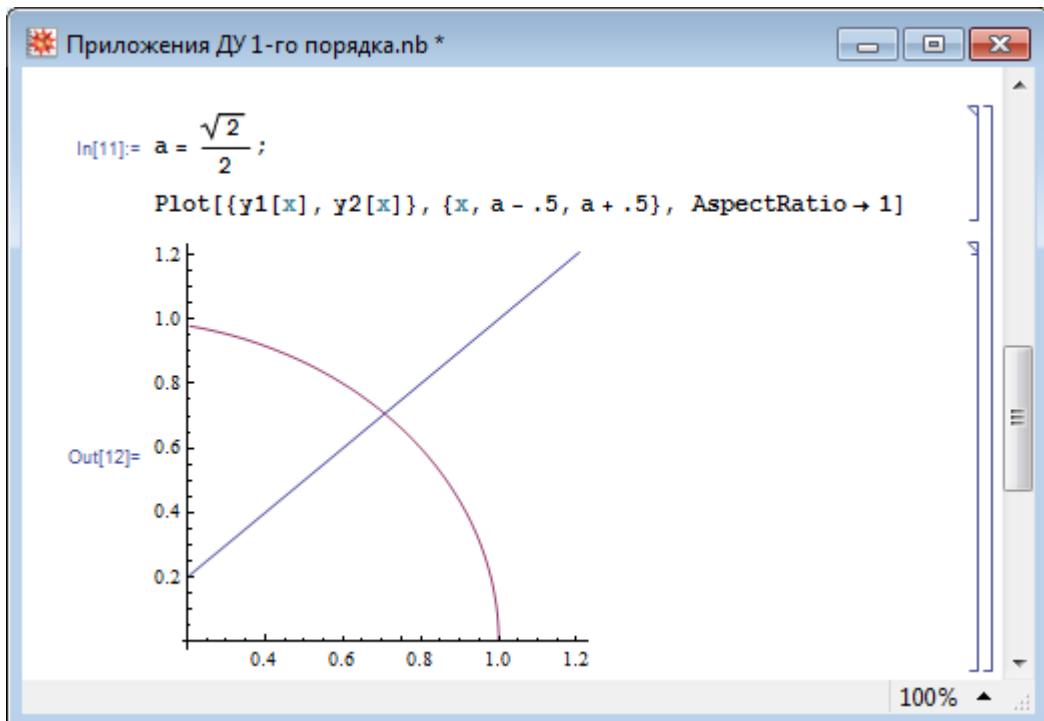
In[9]:= m2 = derivy2 /. x -> \sqrt{2}/2
Out[9]= -1

In[10]:= m1 == -1/m2
Out[10]= True

```

Таким образом, условие ортогональности касательных выполняется и следовательно, кривые ортогональны в данной точке.

Построим данные кривые в некоторой окрестности заданной точки:



Из графика также видно, что кривые пересекаются под прямым углом.

Пример 1.2. Найти семейство ортогональных траекторий для семейства эллипсов $x^2 - xy + y^2 = c^2$. Изобразить на графике эти семейства.

Определим дифференциальное уравнение, соответствующее семейству эллипсов. Продифференцируем обе части уравнения по x . Получим $2x - y - xy' + 2yy' = 0$. Разрешим относительно y' : $y' = \frac{2x - y}{x - 2y}$:

Приложения ДУ 1-го порядка.nb *

```
In[13]:= sol = Solve[D[x^2 - x y[x] + y[x]^2 == 0, x], y'[x]]
```

```
Out[13]= {y'[x] → (2 x - y[x])/(x - 2 y[x])}
```

100%

Следовательно, семейство ортогональных траекторий удовлетворяет уравнению $y' = \frac{2y-x}{2x-y}$:

Приложения ДУ 1-го порядка.nb *

```
In[14]:= orth = -1/sol[[1, 1, 2]]
```

```
Out[14]= -(x - 2 y[x])/(2 x - y[x])
```

100%

Это – однородное уравнение первого порядка, которое может быть решено непосредственно функцией *DSolve*, но получается очень громоздкая формула:

Приложения ДУ 1-го порядка.nb *

```
In[15]:= ds = DSolve[y'[x] == orth, y[x], x]
```

```
Out[15]= {y[x] →
```

$$\frac{-x + \frac{e^{2c[1]}}{3^{1/3} \left(-9 e^{2c[1]} x + \sqrt{3} \sqrt{-e^{6c[1]} + 27 e^{4c[1]} x^2} \right)^{1/3}} + \frac{\left(-9 e^{2c[1]} x + \sqrt{3} \sqrt{-e^{6c[1]} + 27 e^{4c[1]} x^2} \right)^{1/3}}{3^{2/3}}, \{y[x] \rightarrow$$

$$\frac{\left(1 + i \sqrt{3} \right) e^{2c[1]}}{2 3^{1/3} \left(-9 e^{2c[1]} x + \sqrt{3} \sqrt{-e^{6c[1]} + 27 e^{4c[1]} x^2} \right)^{1/3}} - \frac{\left(1 - i \sqrt{3} \right) e^{2c[1]}}{2 3^{1/3} \left(-9 e^{2c[1]} x + \sqrt{3} \sqrt{-e^{6c[1]} + 27 e^{4c[1]} x^2} \right)^{1/3}}\}}$$

100%

Решим это уравнение с помощью замены $y=vx$, где v – новая искомая функция. Уравнение примет следующий вид:

Приложения ДУ 1-го порядка.nb *

```
In[21]:= equ = D[x v[x], x] == Simplify[orth /. y[x] → x v[x]]
```

```
Out[21]= v[x] + x v'[x] = (1 - 2 v[x]) / (-2 + v[x])
```

100%

Разрешим полученное уравнение относительно $v'[x]$ и разделим переменные, поделив на соответствующее выражение:

Найдем интеграл левой и правой частей последнего уравнения отдельно:

Запишем решение уравнения с разделяющимися переменными и сделаем обратную замену:

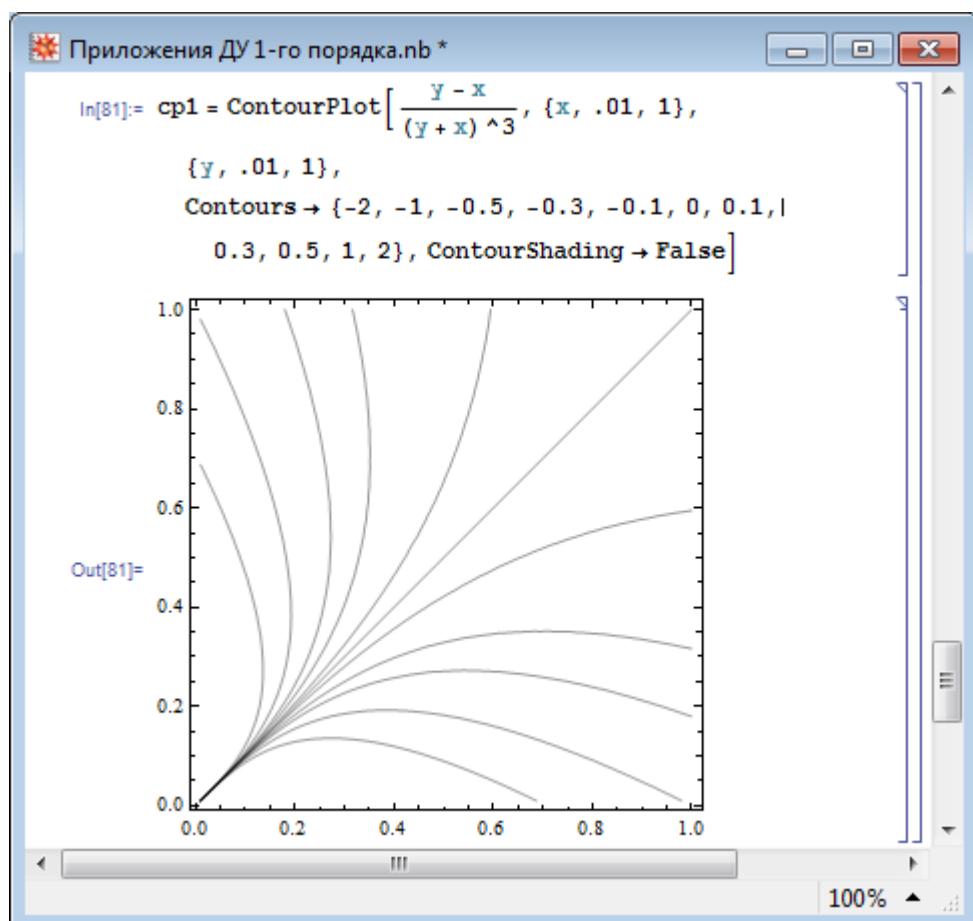
$$\ln\left(\frac{\frac{y}{x}-1}{\left(\frac{y}{x}+1\right)^3}\right) = \ln(x^2) + c,$$

Последнюю формулу можно переписать в виде

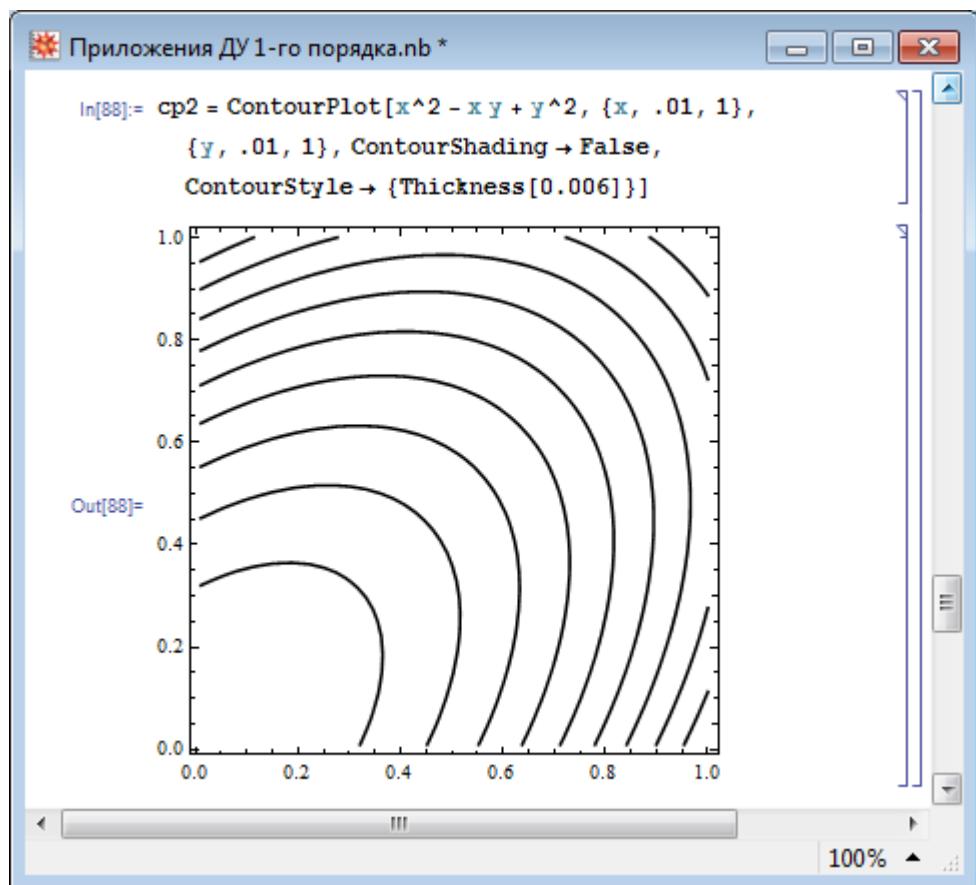
$\frac{\frac{y}{x}-1}{\left(\frac{y}{x}+1\right)^3} = Ax^2$ или $\frac{x^2(y-x)}{(y+x)^3} = Ax^2$. Следовательно, все ортогональные траектории

задаются уравнением $\frac{y-x}{(y+x)^3} = A$. Построим их графики с помощью функции

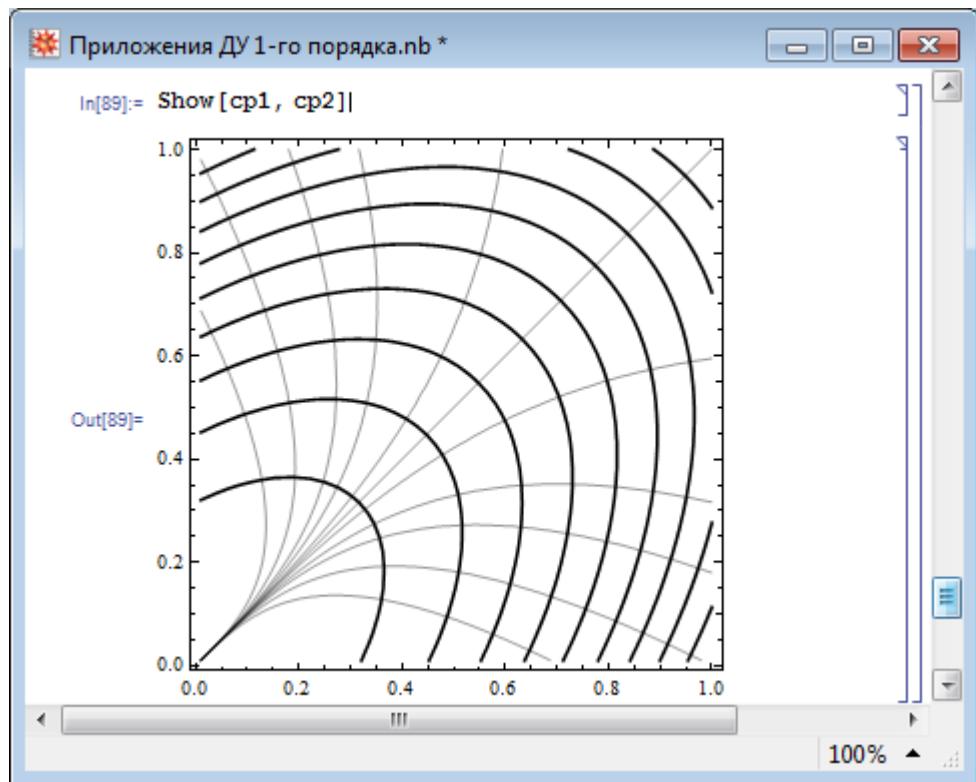
ContourPlot:



Построим также семейство эллипсов:



Отобразим графики на одной координатной плоскости:



§2. ПОЛЕ НАПРАВЛЕНИЙ

Еще один геометрический объект, связанный с дифференциальными уравнениями, это – поле направлений. Рассмотрим уравнение первого порядка, разрешенное относительно производной $y' = f(x,y)$. Правая часть этого уравнения определяет некоторое поле направлений, которое можно получить, если через каждую точку (x,y) , в которой задано уравнение, провести отрезок, образующий с осью Ox угол α , причем $\operatorname{tg}\alpha = f(x,y)$, а в точках, где $f(x,y)$ обращается в бесконечность, направление поля параллельно оси Oy .

Поле направлений и интегральные кривые связаны между собой. Направление касательной в каждой точке интегральной кривой совпадает с направлением поля в этой точке. Это дает возможность по виду поля направлений получить представление об интегральных кривых, то есть о решениях уравнения.

Пример 2.1. Построить поле направлений, связанное с дифференциальным уравнением $y' = e^{-x} - 2y$.

Для построения поля направлений в системе *Mathematica* можно использовать функцию *PlotVectorField*. По умолчанию эта функция не доступна, для ее использования необходимо подключить пакет *VectorFieldPlot*.

Подключим соответствующий пакет. Затем используем команду *PlotVectorField* для построения поля направлений. Первым аргументом этой функции является список, первый элемент которого f , а второй функция $f(x,y)$. Второй и третий аргумент функции определяют область, в которой строится поле направлений:

```
Приложения ДУ 1-го порядка.nb *
In[1]:= << VectorFieldPlots`;

In[2]:= VectorFieldPlot[{1, e^-x - 2 y}, {x, -1, 1},
{y, -1, 1}]

Out[2]=
```

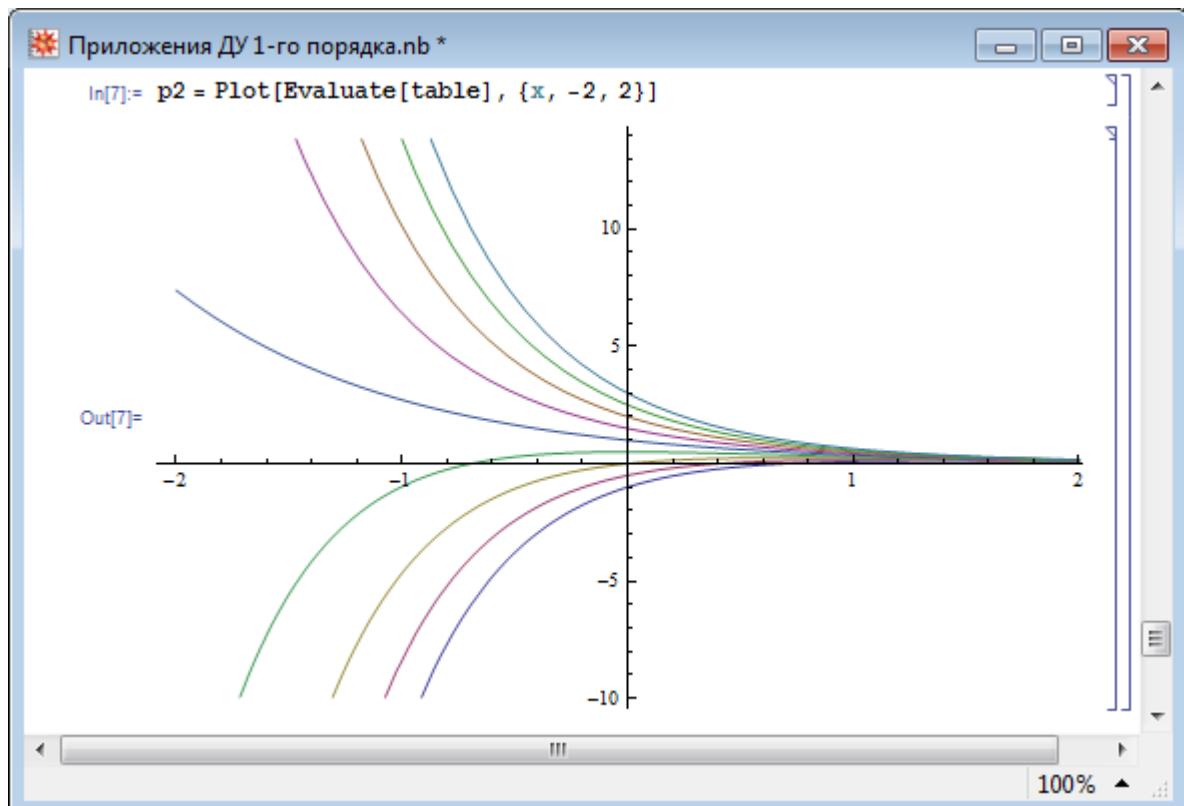
Сравним поле направлений и решение дифференциального уравнения.
Решим с помощью *DSolve* данное дифференциальное уравнение:

```
In[5]:= sol = DSolve[y'[x] == e^-x - 2 y[x], y[x], x]
Out[5]= \{y[x] \rightarrow e^-x + e^-2x C[1]\}
```

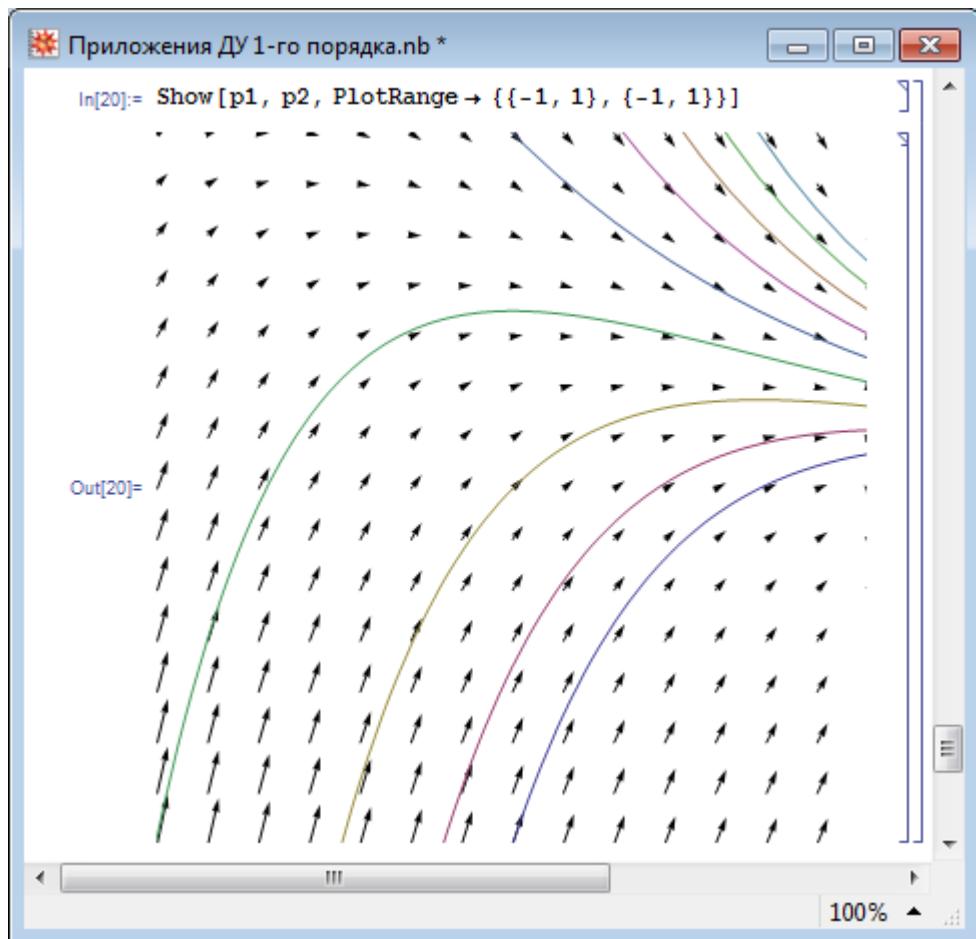
Построим таблицу решений, заменив $C[1]$ на a , где $a=\{-2,-1.5,-1,-0.5,0,0.5,1,1.5,2\}$:

```
In[6]:= table = Table[sol[[1, 1, 2]] /. C[1] \rightarrow a,
{a, -2, 2, .5}]
Out[6]= \{-2. e^-2x + e^-x, -1.5 e^-2x + e^-x, -1. e^-2x + e^-x,
-0.5 e^-2x + e^-x, 0. e^-2x + e^-x, 0.5 e^-2x + e^-x,
1. e^-2x + e^-x, 1.5 e^-2x + e^-x, 2. e^-2x + e^-x\}
```

Построим эти решения:



Отобразим два полученных графика одновременно и покажем, что векторы поля направлений являются касательными к решениям дифференциального уравнения:



§3. ВОЗРАСТАЮЩИЕ И УБЫВАЮЩИЕ ПОПУЛЯЦИИ

Предположим, что численность некоторой популяции $y(t)$ определяется пропорционально ее объему. Математически это соответствует задаче Коши для дифференциального уравнения первого порядка $y'(t)=ky(t)$, $y(0)=y_0$, где y_0 – первоначальная численность популяции. Если $k>0$, то популяция возрастающая; если $k<0$, то популяция убывающая.

Еще одна модель для изучения убывающих популяций определяется уравнением логистики $y'(t)=(r-ay(t))y(t)$, $y(0)=y_0$, где r и a – некоторые постоянные.

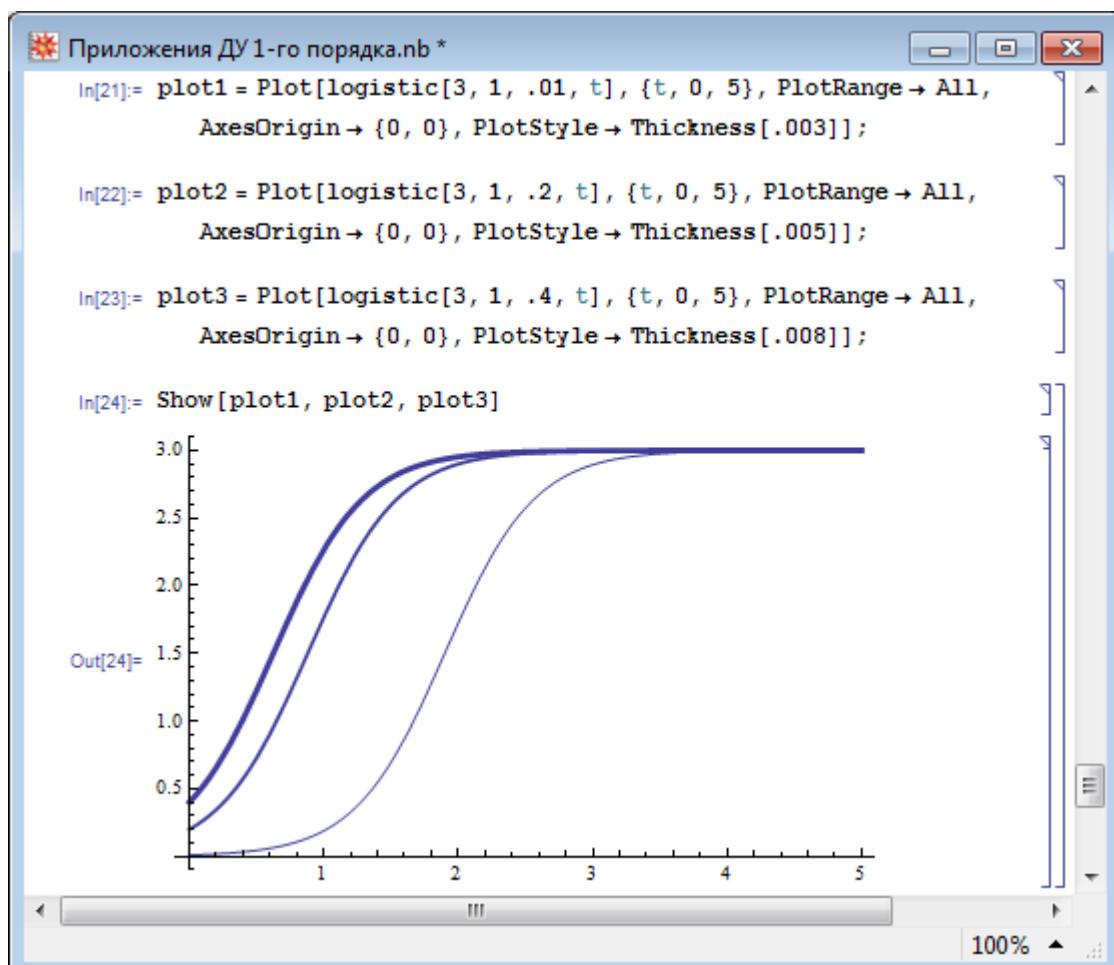
Пример 3.1. Исследовать а) поведение решения в зависимости от первоначальной популяции; б) поведение решения при различных значениях r и a .

Решим уравнение логистики и определим функцию logistic с параметрами r , a , y_0 , и t как решение данного уравнения:

Приложения ДУ 1-го порядка.nb *

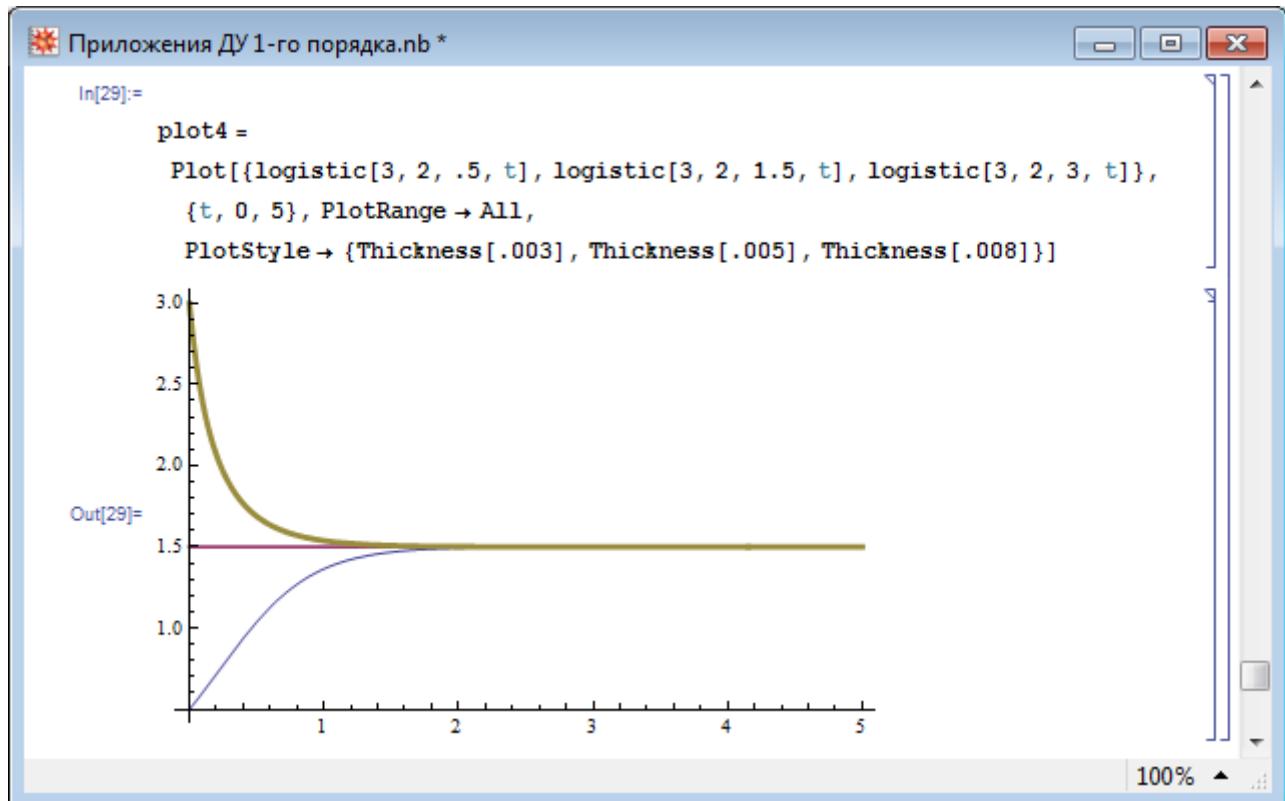
```
In[1]:= det1 = DSolve[{y'[t] == (r - a y[t]) y[t], y[0] == y0},  
    y[t], t]  
  
Out[1]= {y[t] \rightarrow \frac{e^{rt} r y_0}{r - a y_0 + a e^{rt} y_0}}  
  
In[3]:=  
    logistic[r_, a_, y0_, t_] = det1[[1, 1, 2]]  
  
Out[3]= \frac{e^{rt} r y_0}{r - a y_0 + a e^{rt} y_0}
```

Исследуем поведение кривых как зависимость от y_0 для значений 0.01, 0.2 и 0.4 на интервале от 0 до 3. Построим графики для этих значений $plot1$, $plot2$ и $plot3$ соответственно. Значения для r и a возьмем соответственно 3 и 1. Изобразим все графики одновременно:



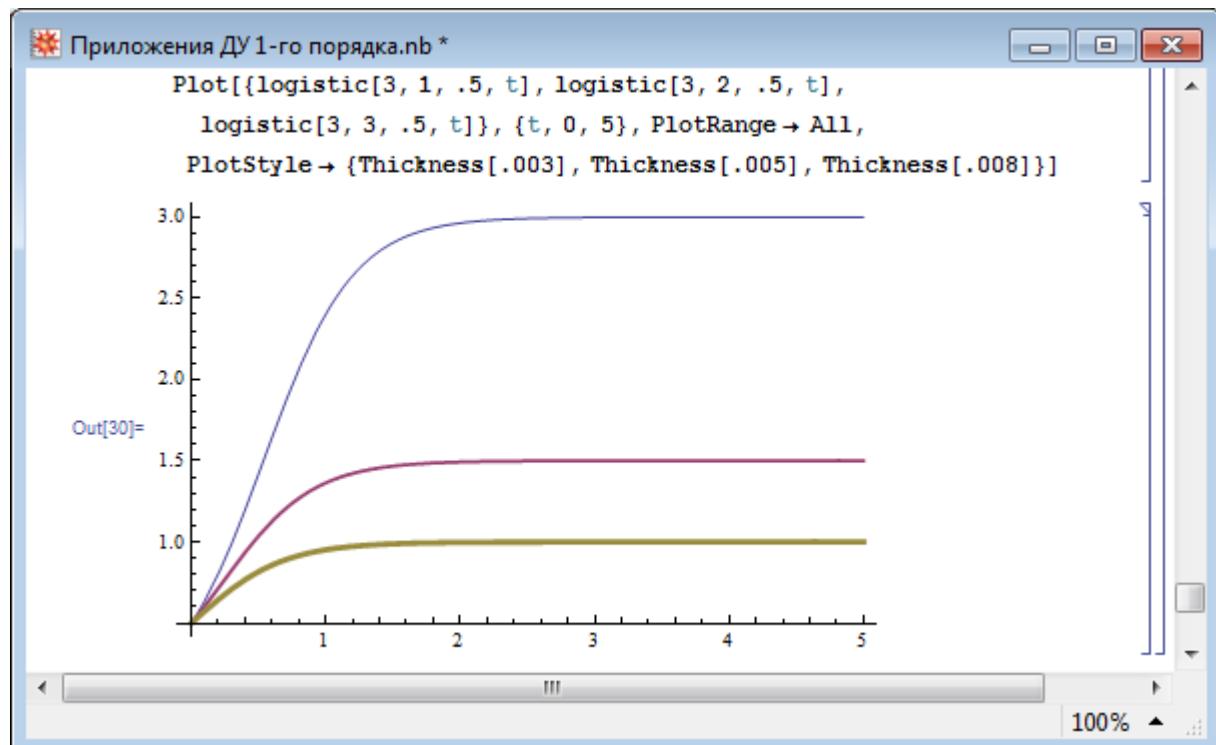
Заметим, что кривые, соответствующие большему значению первоначальной популяции, приближаются к пределу популяции r/a более быстро.

Построим графики, соответствующие значениям $r=3$, $a=2$ для $y_0=0.5$, 1.5 и 3 на одной координатной плоскости:



Заметим, что каждая кривая приближается к предельному значению 1.5 , которое определяется соотношением r/a .

Построим графики, соответствующие значениям $r=3$, $y_0=0.5$ для различных значений $a=1, 2, 3$ на одной координатной плоскости:



Заметим, что каждая кривая приближается к предельному значению r/a , которое равно $3, 1.5, 1$ соответственно.

§4. ЗАКОН НЬЮТОНА-РИХМАНА (О ТЕПЛООБМЕНЕ)

Процесс теплообмена между поверхностью тела и средой описывается законом Ньютона-Рихмана, который гласит, что количество теплоты, передаваемое конвективным теплообменом, прямо пропорционально разности температур поверхности тела $T(t)$ и окружающей среды T_s . Эта ситуация соответствует задаче Коши с дифференциальным уравнением первого порядка $T'(t) = -k(T(t) - T_s)$, $T(0) = T_0$, где T_0 – первоначальная температура тела, а k – коэффициент теплоотдачи, характеризующий интенсивность теплообмена между поверхностью тела и окружающей средой.

Пример 4.1. В расследовании убийств часто бывает важно время смерти. Для определения этого времени может быть использован закон Ньютона-Рихмана. Так, например, нормальная температура тела большинства здоровых людей составляет 98.6 градусов по Фаренгейту. Предположим, что в то время, когда тело было обнаружено, его температура была 82 градуса. Три часа спустя температура снизилась до 75 градусов. Выяснить, сколько времени прошло с момента смерти, если температура окружающей среды 65 градусов?

Решим дифференциальное уравнение этой задачи и определим функцию $temp$ с параметрами $tempS$, $temp0$, k , и t как решение данного уравнения:

```
In[2]:= det1 = DSolve[{TT'[t] == -k (TT[t] - tempS), TT[0] == temp0}, TT[t], t]
Out[2]= {{TT[t] \[Rule] E-kt (temp0 - tempS + Ekt tempS)}}
In[3]:= temp[tempS_, temp0_, k_, t_] = det1[[1, 1, 2]]
Out[3]= E-kt (temp0 - tempS + Ekt tempS)
```

Вычислим коэффициент k с помощью функции $Solve$. Поскольку известно, что при обнаружении температура тела была 82, а через 3 часа – 75 градусов, при температуре окружающей среды 65 градусов:

```
In[4]:= k1 = Solve[temp[65, 82, k, 3] == 75, k] // N
          Solve::ifun :
          Inverse functions are being used by Solve, so some solutions may not
          be found; use Reduce for complete solution information. >>
Out[4]= {k \[Rule] 0.176876}
```

Mathematica выдает сообщение о том, что не все корни найдены, поскольку есть еще комплексные корни, которые не находятся в данном случае. Так как в нашем случае достаточно вещественного корня, то это не влияет на дальнейшее решение задачи.

Построим график решения дифференциального уравнения и вычислим приблизительное время смерти. Поскольку время, когда тело было обнаружено, принято за $t=0$, то график нужно строить для отрицательных значений t :

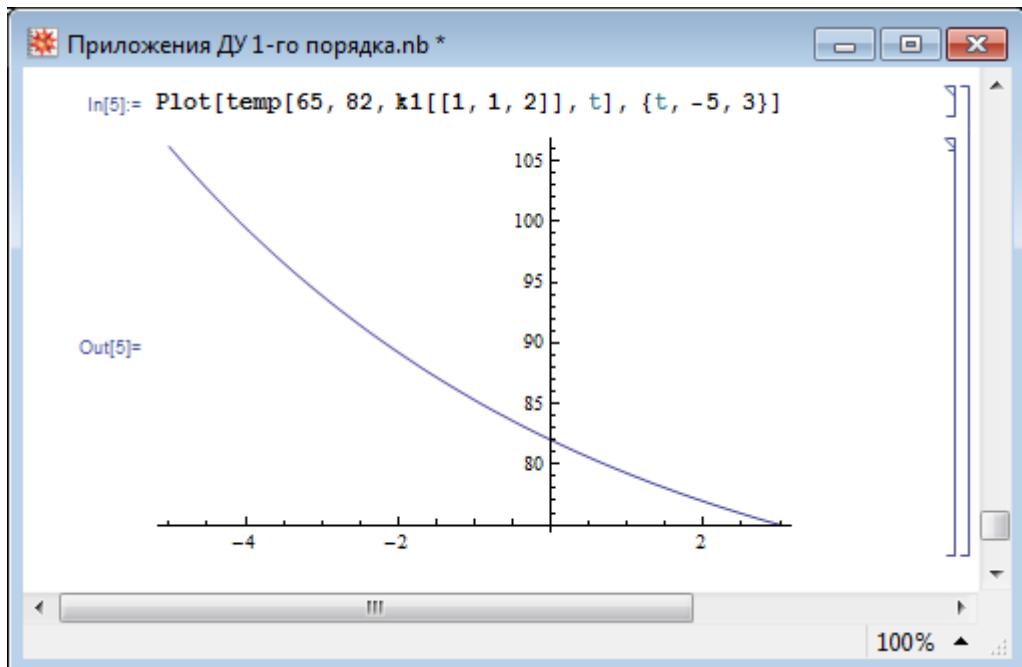
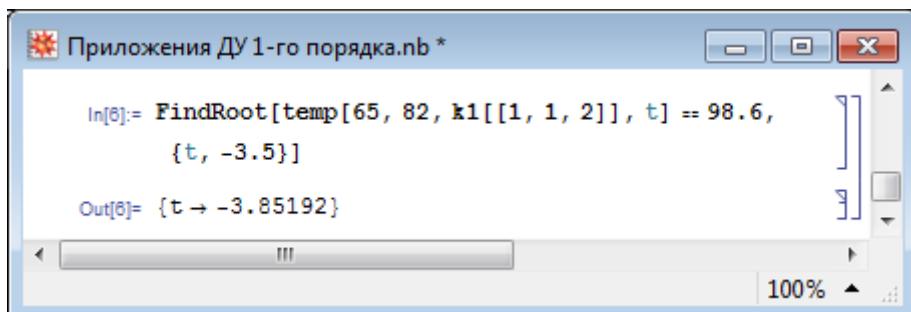


График показывает, что смерть произошла около 3.5 часов назад. Используем функцию *FindRoot*, чтобы получить более точное время смерти:



Вывод: человек умер почти за 4 часа до обнаружения.

Задачи для самостоятельного решения

А. Построить семейства ортогональных кривых к кривым, заданным ниже приведенными уравнениями. Изобразить семейства данных кривых и ортогональных к ним на одном графике:

$$1. \quad x^2 + y^2 - 4x = 0$$

$$2. \quad \frac{x^2}{9} - \frac{y^2}{16} - 1 = 0$$

3. $4x^2 + 5y^2 = 20$
4. $x^2 - 12y = 0$
5. $5x^2 + 4y^2 = 20$
6. $x^2 - 4y^2 + 12x - 4y - 4 = 0$
7. $x^2 + 16y^2 = 16$
8. $y^2 + 8x = 16$

В. Найти общие решения уравнений. Построить поле направлений и интегральные кривые, сравнить их:

1. $y' = y - x^2$
2. $xy' + y = y^2$
3. $y' - 2y = e^x$
4. $xy' = x + y$
5. $2y' - 6y = e^{3x}$
6. $xy' - 2y = 2x^4$
7. $xy = (1+x)y'$
8. $3y' + 12y = e^{-x}$

С. Решить следующие задачи. Результаты изобразить графически:

1. Численность населения страны S была рекордной 5.3 млн. в 1800 году. Определить актуальную на данный момент численность населения, если коэффициент пропорциональности численности популяции к ее объему $k=0.03$.
2. Тело охладилось за 10 минут от 100 до 60°C . Температура окружающего воздуха поддерживается равной 20°C . Когда тело остынет до 25°C ?
3. За 30 дней распалось 50% первоначального количества радиоактивного вещества. Через сколько времени останется 1% от первоначального количества?
4. Температура только что испекшегося пирога составляет 180°C . Через 15 минут его температура составляет 66°C . Определите, через какое время пирог остынет до 27°C , чтобы его можно было есть.
5. Изначальная масса радиоактивного вещества 10 грамм. Период его полураспада составляет 1 год. Определить, через какой промежуток времени останется 9 грамм вещества. Сколько грамм вещества останется через 6 лет?
6. Для длительного хранения ягоды поместили в морозильную камеру с температурой -17°C . Через 5 минут их температура составляет 20°C .

Определить, через какое время ягоды заморозятся (0°C), если их первоначальная температура 24°C ?

7. Исследовать уравнение логистики при различных значениях начальной популяции и фиксированных коэффициентов r и a . В каком случае наблюдается прирост численности популяции, а в каком – снижение?

ЛИТЕРАТУРА

1. *Воробыёв Е. М.* Введение в систему *Mathematica*. М.: Финансы и статистика, 1997.
2. *Дьяконов В. П.* *Mathematica 5.1/5.2/6. Программирование и математические вычисления*. М.: ДМК-Пресс, 2008.
3. *Прокопеня А. Н., Чичурин А. В.* Применение системы *Mathematica* к решению обыкновенных дифференциальных уравнений. Минск: БГУ, 1999.
4. *Abel M. L., Braselton J. P.* Differential equations with *Mathematica*. London: Academic Press Inc, 1993.
5. *Dubin D.* Numerical and analytical methods for scientists and engineers using *Mathematica*. New Jersey: John Wiley & Sons Inc, 2003.
6. *Wellin P., Kamin S., Gaylord R.* An introduction to programming with *Mathematica*. NY: Cambridge University Press, 2005.
7. *Эльсгольц Л. Э.* Дифференциальные уравнения: учебник для физических и физико-математических факультетов университетов. М.: URSS, 2013.
8. *Филиппов А. Ф.* Сборник задач по дифференциальным уравнениям. Ижевск: РХД, 2005.