

§2. Основы работы в системе Wolfram Mathematica

Запуск программы. Главное меню программы.

Для запуска программы **Mathematica** необходимо щелкнуть иконку **Mathematica** в меню «Программы» или ярлык программы в месте его расположения.

При запуске программы на экране появляется главное окно (рис. 2.1).

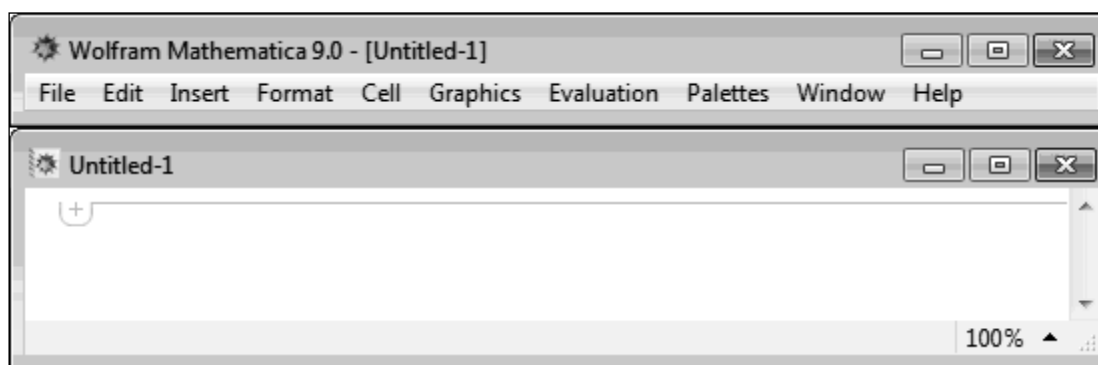


Рис. 2.1

Главное окно системы содержит строку заголовка, главное меню (File, Edit, Insert, Format, Cell, Graphics, Evaluation, Palettes, Window, Help) и большой экран редактирования (окно ввода).

Краткое описание пунктов главного меню системы.

- 1) **File** – действия с файлами программы:
 - | **New** – создание нового файла;
 - | **Open** – открытие из каталога уже созданного файла;
 - | **Close** – закрытие файла без сохранения;
 - | **Save** – сохранение файла с прежним именем;
 - | **Save As** – сохранение файла с новым именем;
 - | **Printing Settings** – управление параметрами представления данных на экране;
 - | **Print** – печать документа;
 - | **Exit** – завершение работы всей программы.
- 2) **Edit** – операции редактирования:
 - | **Undo** – отмена операции;
 - | **Cut** – удаление выделенного фрагмента документа и помещение его в буфер обмена;
 - | **Copy** – копирование;
 - | **Paste** – вставка фрагментов из буфера обмена в заданную область документа;

- | **Clear** – удаление фрагментов документа без его сохранения в буфере;
- | **Copy As** – копирование в заданном формате;
- | **Select All** – выделение всего документа;
- | **Check Spelling** – проверка орфографии;
- | **Find** – задание шаблона для поиска;
- | **Find Next** – переход к следующему фрагменту, совпадающему с шаблоном;
- | **Find Previous** – переход к предыдущему фрагменту, совпадающему с шаблоном;
- | **Preferences** – вызов окна настроек системы.

3) **Insert** – введение данных в окно редактирования. Например, в подпункте **Typesetting**:

- | **Superscript** – верхний индекс;
- | **Subscript** – нижний индекс;
- | **Matching []** – текст в скобках [];
- | **Matching ()** – текст в скобках ();
- | **Matching { }** – текст в скобках { } и т. д.

Можно вставлять в текст графику, формулы и т. д.

4) **Format** – изменение формата текста на экране и при печати, установка стилей, управление окном редактирования, стиль ячеек, их содержание, размер, управление шрифтами и т. д.:

- | **Style** – установка параметров текста (шрифт, размер символов и т.д.);
- | **Screen Environment** – изменение формата текста на экране; имеет следующие установки:

-) **Working** – стиль типичный;
-) **Presentation** – презентационный стиль с увеличением размера символов;
-) **Condensed** – сжатый размер символов;
-) **Printout** – оптимальный стиль для печати.

5) **Cell** – работа с ячейками:

- | **Convert To** – преобразование формата содержимого ячеек:
 -) **InputForm** – формат ввода;
 -) **OutputForm** – формат вывода;
 -) **StandartForm** – стандартный формат;
 -) **TradicionalForm** – традиционный формат;
 -) **Bitmap** – растровый формат изображений.

При работе с большим числом математических знаков целесообразно использовать стандартный формат.

- └ **Cell Properties** – установление формата ячеек:

- └ **Open** – устанавливает ячейку открытой или закрытой;

- └ **Editable** – устанавливает ячейку редактируемой или нередактируемой;

- └ **Evaluatable** – устанавливает ячейку оцениваемой или неоцениваемой;

- └ **Initialization Cell** – делает ячейку инициализационной или неинициализационной.

- └ **Grouping** – группировка ячеек:

- └ **Automatic Grouping** – объединение ячеек в соответствии с их стилем;

- └ **Manual Grouping** – объединение и разъединение ячеек.

По умолчанию выбран режим **Automatic Grouping**.

- └ **Divide Cell** – разделение сгруппированных ячеек;

- └ **Merge Cells** – объединение выделенных ячеек.

6) **Graphics** – работа с графическими данными:

- └ **New Graphics** – вывод окна для построения графика;

- └ **Drawing Tools** – вывод окна графического редактора;

- └ **Rendering** – вывод подменю операций рендеринга:

- └ **Animate Selected Graphics** – анимация с графиком выделенной ячейки;

- └ **Align Selected Graphics** – выравнивание графиков;

- └ **Make Standart Size** – установка стандартного размера ячейки;

- └ **Rerended Graphics** – построение графиков заново.

7) **Evaluation** – управление процессом вычислений:

- └ **Evaluate Cells** – вычисление выделенных ячеек;

- └ **Evaluate in Place** – вычисление выделенных выражений в строке ввода;

- └ **Evaluate Initialization Cells** – вычисление инициализированных ячеек без их выделения;

- └ **Evaluate Notebook** – вычисление всех ячеек документа;

- └ **Interrupt Evaluation** – прерывание текущего вычисления;

- └ **Abort Evaluation** – сбрасывание текущего вычисления;

- └ **Remove from Evaluation Queue** – отмена вычисления ячеек, стоящих на «очереди».

Следующие опции связаны с возможностью использования ядра (**Kernel**) другого компьютера:

- └ **Default Kernel** – выбор ядра, используемого по умолчанию;

- | **Notebook's Kernel** – выбор ядра для вычислений в текущем документе;
- | **Start Kernel** – запуск выбранного ядра;
- | **Quit Kernel** – завершение работы ядра.

8) **Palettes** – управление вводом данных.

Mathematica позволяет осуществлять ввод данных в окно ввода двумя способами: вручную с клавиатуры и с использованием так называемых палитр (**Palettes** – панели с кнопками быстрого управления). Они представляют собой окна, содержащие набор кнопок, за которыми закреплены определенные действия, и выпадающих списков (рис. 2.2). Палитры можно выводить на экран и убирать с экрана, создавать собственные палитры с требуемым набором функций.

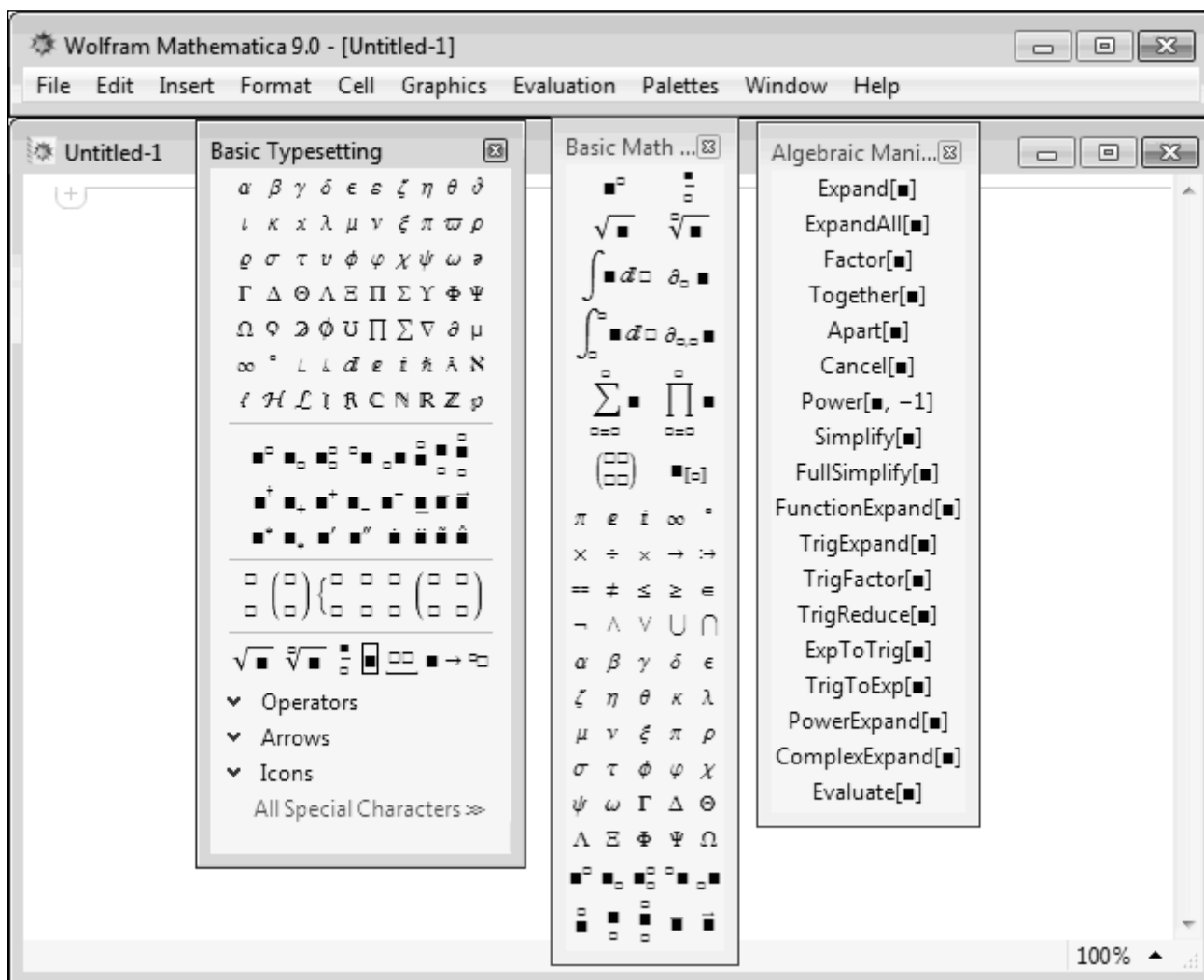


Рис. 2.2

9) **Window** – управление внешним видом окон:

- | **Stack Windows** – каскадное расположение окон;
- | **Tile Windows Wide** – расположить на экране окна открытых документов одно над другим вытянутыми по ширине;

- | **Tile Windows Tall** – расположить на экране окна открытых документов одно рядом с другим вытянутыми по длине;
- | **Messages** – вывод окна сообщений об ошибках.

10) **Help** – управление справочной системой.

В любом из пунктов главного меню некоторые команды могут быть выделены светло-серым шрифтом. Это означает, что команды не могут быть выполнены в данный момент. Например, если выражение не выделено, то его значение вычислить нельзя.

Окно ввода (экран редактирования).

На экране появляется активное окно документа. По умолчанию создаваемый документ носит название **Untitled-1**. При сохранении можно присвоить ему нужное имя. Система автоматически присваивает файлам расширение **.nb**.

В программе **Mathematica** все введенные в окно ввода данные содержатся в отдельных, определенным образом выделенных областях экрана, называемых *ячейками*. Введенные данные автоматически объединяются во входную ячейку, которая обозначается квадратной скобкой –] в правой части окна ввода. Например, наберем **3 + 7** (рис. 2.3).

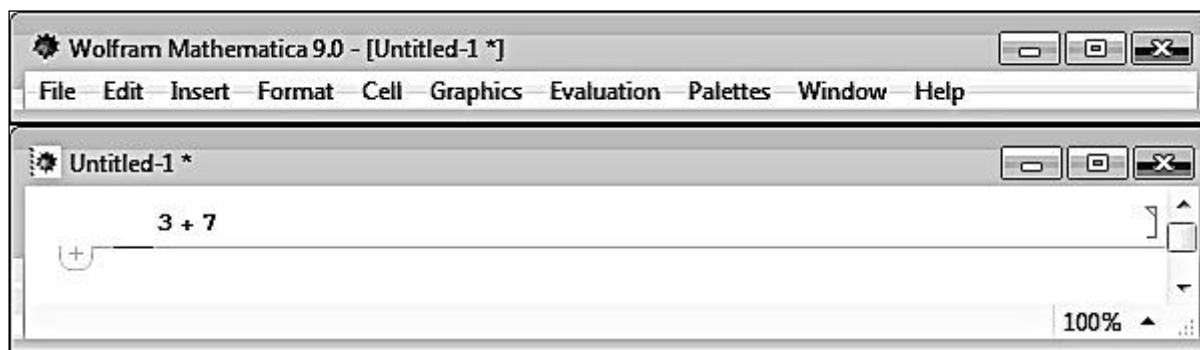


Рис. 2.3

Для получения результата нужно поместить курсор в любой части ячейки и нажать **Shift + Enter** (удерживая **Shift**, нажать **Enter**). Нажатие одной клавиши **Enter** приводит к созданию новой строки в той же ячейке.

Если введенные данные являются логически завершенными и не содержат синтаксических ошибок, программа **Mathematica** обрабатывает их и выдает результат. В противном случае указывается тип ошибки.

Результат **3 + 7** на рис. 2.4.

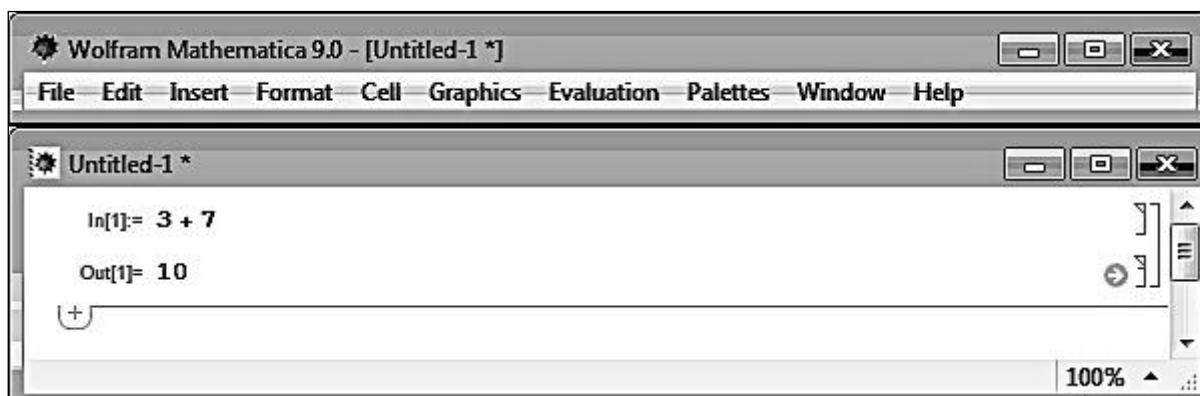


Рис. 2.4

На рис. 2.4 **Mathematica** добавляет к данным на экране метки:

- 1) **In[n]:=** – вводимые пользователем данные;
- 2) **Out[n]:=** – результат, выводимый программой **Mathematica**, где **n=1, 2, 3 ...** – номер проводимого вычисления. Номер **n** может быть использован *для ссылки на результат выполнения n-й операции* с помощью записи **%n**.

Входную и выходную ячейки окаймляют квадратные скобки, а вместе они ограничены общей квадратной скобкой – это значит, что сформирована группа ячеек.

Для окончания работы с пакетом **Mathematica** необходимо выбрать команду **Exit** в разделе **File** главного меню. Если требуется сохранить введенные данные, то появляется дополнительное окно, в котором можно определить имя сохраняемого документа. Этот файл можно открыть при следующем сеансе работы с программой **Mathematica**, выбрав в разделе **File** команду **Open**.

Некоторые встроенные функции системы Mathematica.

Названия всех встроенных функций системы **Mathematica** записываются с большой буквы. Если название функции состоит нескольких «склеенных» слов, то, как правило, каждое из них записывается с большой буквы. Аргументы всех функций, как встроенных, так и задаваемых пользователем, всегда заключаются в **квадратные скобки**. Уравнения в системе **Mathematica** формируются двойным знаком равно, т.е. «**==**».

Abs[x] – абсолютная величина (модуль) x .

And[a,b,c,...] – логическое умножение или логическое «И». Дает значение *True*, если логические значения a, b, c, \dots равны, в противном случае дает значение *False*.

Append[*lst*,*x*] – создает новый список, добавляя элемент *x* в конец списка *lst*.

Array [*a*,*n*,*k*] – символьный список $\{a[k], a[k+1], \dots, a[k+n-1]\}$, состоящий из *n* элементов. Если $n=0$, то функция дает пустой список $\{\}$. Аргумент *k* (начальное значение индекса) может быть нулевым или отрицательным.

Array[*a*,*n*] эквивалентно **Array**[*a*,*n*,1], то есть дает список $\{a[1], a[2], \dots, a[n]\}$.

Array[*a*,{*n*,*m*}] – дает сложный список $\{\{a[1,1], a[1,2], \dots, a[1,m]\}, \{a[2,1], a[2,2], \dots, a[2,m]\}, \dots, \{a[n,1], \dots, a[n,m]\}\}$, состоящий из *n* списков длины *m*.

Clear[*s1*,*s2*,...] – стирает любые значения, присвоенные указанным символам *s1*, *s2*,...

Collect[*expr*,*x*] – группирует члены выражения *expr* с одной и той же степенью переменной *x*.

ColumnForm [*lst*,*w1*,*w2*] – выводит список *lst* на экран в виде колонки. Аргументы *w1* и *w2* необязательны. Первый (*w1*) задает способ выравнивания элементов списка в колонке по горизонтали:

Center–по центру, *Left*–по левому краю, *Right*–по правому краю.

Второй (*w2*) определяет выравнивание в ячейке вывода по вертикали:

Center–по центру, *Below*–по верхнему краю, *Above*–по нижнему краю.

ColumnForm[*lst*] эквивалентно **ColumnForm**[*lst*,*Left*,*Below*].

Exp[*x*] – e^x , где *e* – основание натурального логарифма.

expr/.x→value – выполняет подстановку значения *value* вместо *x* в выражение *expr*, которое зависит от *x*.

Floor[*x*] – наибольшее целое число, не превосходящее *x*.

For[*start*,*test*,*incr*,*body*] – цикл с параметром. Выражения *start* и *body* могут состоять из нескольких операторов, которые разделяются знаком «;» (точка с запятой). Сначала один раз вычисляется выражение *start* (начальные действия). Если справедливо условие *test*, зависящее от параметра цикла, то вычисляются выражения *body* (тело цикла) и *incr* (приращение параметра), и все повторяется. Тело цикла может ни разу не выполниться. В цикле **For** нельзя задать локальные переменные.

FractionalPart[*x*] – дробная часть числа *x*.

If[*test*,*expr1*,*expr2*] – определяет выбор и вычисление выражения. Сначала вычисляется логическое выражение *test*. Если получено значение *True*, то вычисляется выражение *expr1*, в противном случае вычисляется выражение *expr2*.

IntegerPart[*x*] – целая часть числа *x*.

Graphics[{*drvs*,*prvs*},*optns*] – формирует двумерные графические объекты-примитивы *prvs* (круги, диски, линии, прямоугольники, многоугольники, текст), которые с помощью функции *Show* можно показать на одном графике. Директивы *drvs* определяют размеры, цвет, стиль оформления рисуемых линий и текста. Они ставятся перед примитивами и могут действовать на несколько из них.

Для функций *Graphics* и *Plot* можно задавать одни и те же опции *optns*.

GraphicsArray[*lst*] – позволяет разместить в одном графическом окне на экране сразу несколько графиков, возможно разных системах координат. Графики можно расположить в одну строку или в виде таблицы. Здесь *lst* – линейный или двухуровневый список, который группирует графические объекты.

Length [*lst*] – длина (количество элементов) списка *lst*.

List[*a*,*b*,*c*,...] – формирует список {*a*,*b*,*c*,...}. Элементы *a*,*b*,*c*,... в свою очередь могут быть списками, а также любыми другими выражениями. **List**[] дает пустой список, то есть {}.

ListPlot [*lst*,*optns*] –предназначена для построения графика по точкам. В общем случае аргумент *lst* представлен списком {{*x1*,*y1*},{*x2*,*y2*},...}, где (*x1*,*y1*),(*x2*,*y2*),...–координаты отмечаемых точек. Список {{1,*y1*}, {2,*y2*},...} можно задать как {*y1*,*y2*,...}. Список также можно создать какой-либо функцией, например, *Range* или *Table*.

Для функции *ListPlot* используются, в основном, те же опции *optns*, что и для функции *Plot*. Если требуется соединять точки на графике отрезками прямых, то нужно установить дополнительную опцию *PlotJoined*→*True*. Для управления размером точек используется директива *PointSize*, а не *Thickness*, как для линий.

Log[*a*,*x*] – логарифм *x* по основанию *a*. **Log**[*x*] – натуральный логарифм *x*.

MatrixForm[*lst*] – выводит на экран двухуровневый список *lst* в виде матрицы, имеющей ячейки одного и того же размера. Линейный список выводится в виде столбца.

Max [*a,b,c,...*] – максимальное значение из *a,b,c,...*. Любой из аргументов может быть списком.

Min [*a,b,c,...*] – минимальное значение из *a,b,c,...*. Любой из аргументов может быть списком.

Mod [*m,n*] – остаток от деления целого *m* на целое *n*.

N[*expr,n*] – вещественное значение выражения *expr*, *n*–разрядность, используемая при вычислении результата. Если второй аргумент не задан, то разрядность выбирается автоматически, а результат выводится с шестью значащими цифрами.

NestList [*f,x,n*] – формирует список результатов *n*-кратного применения функции *f* к аргументу *x*, то есть список $\{x, f(x), f(f(x)), \dots\}$.

PaddedForm[*expr,{m,n}*] – задает размер *m* (количество цифр) десятичного представления вещественного значения выражения *expr* при выводе его на экран. Здесь *n*–количество цифр после десятичной точки.

Plot[$\{f_1, f_2, \dots\}, \{x, x_{min}, x_{max}\}, optns$] – предназначена для построения графиков функций $y = f_1(x)$, $y = f_2(x)$, ... при изменении независимой переменной *x* в пределах от *xmin* до *xmax*. При этом используется прямоугольная (декартова) система координат. Необязательные аргументы *optns* (опции), общие и для других графических функций, служат для настройки вида графиков. Если опции не указаны, то их стандартные значения устанавливаются автоматически.

Prepend[*lst,x*] – создает новый список, добавляя *x* в начало списка *lst*.

Print[*expr1,expr2,...*] – выводит на экран значения указанных выражений *expr1,expr2,...*. Следующий вывод переносится в начало новой строки.

Random[] – при последовательных вызовах дает случайные вещественные числа, равномерно распределенные на интервале от 0 до 1.

Range[*m, n,d*] – формирует числовой список, значения которого равномерно распределены в интервале от *m* до *n* с шагом *d*. Значения *m, n, d* – обязательно целые.

Rationalize[*x*] и **Rationalize**[*x, точностьчисла x*] – представляет *x* в виде рационального числа $\frac{m}{n}$ (с заданной точностью).

Return[*expr*] – вызывает завершение выполнения подпрограммы-функции. Выражение *expr* определяет значение, возвращаемое как результат. Часто рассматривается как дополнительная, возможно, для исключительных ситуаций, точка завершения подпрограмм.

Show[*plot,opns*] – выводит на экран уже сформированный (например, функцией *plot*) график. С помощью второго параметра можно изменить значения опций в той графической функции, при помощи которой был получен рисунок.

Show[{*plot1,plot2,...*},*opns*] – совмещает в одном графическом окне несколько графиков. Функция полезна в тех случаях, когда желательно, не вычисляя заново исходные графики *plot1, plot2,...* просмотреть их при иных настройках опций *opns* или опоставить.

Simplify[*expr,optns*] – приводит выражение *expr* к простейшей форме. При этом тригонометрические тождества не используются, если указана опция *Trig*→ *False*.

Table[*expr,n*] – список из *n* значений одного и того же выражения *expr*.

Table[*expr,{i,m,n,d}*] – список значений выражения *expr*, зависящего от параметра *i*, для *i* от *m* до *n* с шагом *d*.

Table[*expr,{i,m,n}*] эквивалентно **Table**[*expr,{i,m,n,1}*].

Table [*expr,{i,n}*] эквивалентно **Table**[*expr,{i,1,n,1}*].

Table[*expr,{i,m1,n1},{j,m2,n2},...*] – порождает многоуровневые списки, используется для создания числовых таблиц.

TableForm[*lst,opns*] – выводит на экран двухуровневый список *lst* в виде таблицы, высота строк и ширина столбцов которой определяются максимальными размерами элементов списка. Линейный список представляется строкой или колонкой в зависимости от значения (*Row* или *Column*) опции *TableDirections*. Если установить опцию *TableHeadings*, то можно вывести названия для строк и столбцов.

Timing[*expr*] – значение выражения *expr* и время, затраченное на его вычисление.

While[*test,body*] – цикл с условием. Выражение *body* (тело цикла) записывается в виде последовательности операторов, разделенных знаком «;» (точка с запятой). Сначала вычисляется логическое выражение *test*. Если получено значение *True*, то вычисляется тело цикла, и все повторяется. Если получено зна-

чение *False*, то цикл завершается. Цикл может оказаться бесконечным, если условие *test* никогда не выполняется. Тело цикла может ни разу не вычисляться, так как все начинается с проверки условия. В цикле *While* нельзя задать локальные переменные.

Основные опции графических функций.

В функциях, формирующих графики, в числе других аргументов могут указываться опции, которые записываются в виде подстановок, например, *AspectRatio* → *Automatic*. Они могут иметь числовые или символьные значения, а также значения в виде списков. При желании опции могут быть изменены при вызове функции *Show*. Наиболее распространенные символьные значения опций: *Automatic* – используется автоматический выбор стандартного значения, *None*, *False* – опция не используется, *All* – используется в любом случае.

Директивы, используемые в графических функциях.

Директивы передают значения многим графическим опциям и являются важным средством настройки графиков.

AbsoluteDashing[$\{d_1, d_2, \dots\}$] – устанавливает стиль рисования линии графика пунктиром, т.е. в виде последовательности отрезков длины d_1, d_2, \dots , которые повторяются циклически. Здесь d_1 – длина рисуемой линии, d_2 – пропуск и т.д. Значения задаются в пунктах ($pt = 1/72 \text{ inch} \approx 0,035 \text{ см}$).

AbsolutePointSize[d] – задает диаметр d (в пунктах (pt)) отдельных точек графиков, построенных по точкам.

AbsoluteThickness[d] – задает толщину d линии графика в пунктах (pt).

GrayLevel[d] – задает уровень интенсивности оттенка серого цвета для линий графиков, $0 \leq d \leq 1$ (0 – черный цвет, 1 – белый).

RGBColor[d_1, d_2, d_3] – задает цвет линии графика в цветовой модели *RGB* (красный, зеленый, голубой). Здесь $0 \leq d_1 \leq 1$, $0 \leq d_2 \leq 1$, $0 \leq d_3 \leq 1$ – интенсивности базовых цветов ($1, 0, 0$ – красный цвет; $0, 1, 0$ – зеленый; $0, 0, 1$ – голубой). Директива используется при выводе графиков на экран дисплея.

Простейшие операции со списками в системе Mathematica.

В Mathematica *списки* (*lists*) являются основным способом объединения объектов. Элементы списков всегда заключаются в **фигурные скобки**. Напри-

мер, $\{1,2,3\}$ – список чисел. Списки не выполняют никаких действий над объектами, это лишь способ хранить их как единое целое. Списки $\{a, b, c\}$ и $\{b, c, a\}$ являются различными из-за порядка элементов.

Для выделения элементов списка *list* используются **двойные квадратные скобки**: *list*[[i]] – возвращает i-й элемент списка, *list*[[i,j]] – возвращает элемент двумерного списка (т.е. списка списков), находящийся на позиции (i, j).

Count[list, pattern] – количество элементов списка *list*, соответствующих образцу *pattern*.

Length[list] – число элементов одномерного списка *list* или число размерностей многомерного списка.

Drop[list, n] – удаляет первые *n* элементов списка *list*.

Drop[list, -n] – удаляет последние *n* элементов списка *list*.

Drop[list, {n}] – удаляет *n*-й элемент списка *list*.

Drop[list, {n,m}] – удаляет элементы списка *list* с *n*-го по *m*-й.

Delete[list, n] – удаляет *n*-й элемент списка *list*.

First[list] – первый элемент списка *list*.

Last[list] – последний элемент списка *list*.

Take[list, n] – возвращает первые *n* элементов списка *list*.

Take[list, -n] – возвращает последние *n* элементов списка *list*.

Take[list, {n,m}] – возвращает элементы списка *list* с порядковыми номерами от *n* до *m*.

Rest[list] – возвращает список *list* без первого элемента.

Append[list, element] – добавляет элемент *element* в конец списка *list*.

Prepend[list, element] – добавляет элемент *element* в начало списка *list*.

Insert[list, element, n] – вставляет элемент *element* в позицию *n* списка *list*.

Flatten[list] – превращает список *list* в одномерный (выравнивает список).

Sort[list] – сортирует элементы списка *list* в каноническом порядке.

Reverse[list] – изменяет порядок элементов списка *list* на обратный.

Join[list1, list2,...] – объединяет списки *list1*, *list2*,... в единую цепочку.

Intersection[list1, list2,...] – возвращает упорядоченный список элементов, общих для всех списков *list1*, *list2*,... (т.е. находит пересечение множеств).

Union[list] – сортирует список *list*, удаляя из него все повторяющиеся элементы.

Union [list1, list2,...] – возвращает отсортированный список из всех элементов списков *list1*, *list2*,..., в котором удалены все повторяющиеся элементы (т.е. находит объединение множеств).

Примеры работы со встроенными функциями системы Mathematica, построение таблицы значений и вывод графика функции.

Пример 2.1. Найти: 1) значение выражения $(32 + 48)/11$, выделить его целую и дробную части;
2) значение выражения $\ln(6^5 - 3^8)$ и рациональное приближение к нему с точностью 10^{-4} .

Δ 1) Введем выражение, используя палитру инструментов (**Palettes**→**Basic Math Assistant**) и найдем его значение (**Shift+Enter**).

```
In[1]:=  $\frac{32 + 48}{11}$ 
Out[1]=  $\frac{80}{11}$ 
```

Как видно, результат представлен в виде дроби (рационального числа). Получим численное приближение к значению в виде десятичной дроби с количеством цифр, принятым по умолчанию, а также равным 9.

```
In[2]:= N[ $\frac{32 + 48}{11}$ ]
|численное прибли
```

```
N[ $\frac{32 + 48}{11}$ , 9]
|численное прибли
```

```
Out[2]= 7.27273
```

```
Out[3]= 7.27272727
```

Выделим целую и дробную части числа, сославшись на него, как на результат выполнения первой операции.

```
In[4]:= IntegerPart[%1]
|целая часть
```

```
Out[4]= 7
```

```
In[5]:= FractionalPart[%1]
|дробная часть
```

```
Out[5]=  $\frac{3}{11}$ 
```

```
In[6]:= FractionalPart[%1] // N
|дробная часть |численное приближение
```

```
Out[6]= 0.272727
```

Найдем рациональное приближение к числу, являющемуся результатом третьей операции.

```
In[7]:= Rationalize[%3]
|найти рациональное приближение
```

```
Out[7]= 80
        11
```

2) Найдем числовое значение выражения $\ln(6^5 - 3^8)$ и рациональное приближение к нему с заданной точностью.

```
In[8]:= Log[6^5 - 3^8] // N
|натуральный ло... |численное приближение
```

```
Out[8]= 7.1025
```

```
In[9]:= Rationalize[Log[6^5 - 3^8], 10^-4]
|найти рацио... |натуральный логарифм
```

```
Out[9]= 277
        39
```



Пример 2.2. Задать функцию $f(x) = (x + 1)(x^2 - 4x + 4)$. Раскрыть скобки в определяющем ее выражении, а затем разложить полученный многочлен на множители. Найти значения функции в точках $1,8$; $-e$; $\sqrt[3]{14}$. Найти $f(2a - b)$ и упростить полученное выражение. Составить таблицу из координат точек графика функции на отрезке $[-3, 4]$ с шагом 1 и 0,5 по x . Построить график функции на отрезке $[-3, 4]$.

Δ Зададим функцию $f(x)$. Знак «_» после переменной x применяется для того, чтобы задать x как локальную переменную внутри функции.

```
In[1]:= f[_x_] = (x + 1) {x^2 - 4 x + 4}
```

```
Out[1]= (1 + x) {4 - 4 x + x^2}
```

Раскроем скобки, а затем разложим на множители данную функцию.

```
In[2]:= Expand[f[x]]
|раскрыть скобки
```

```
Out[2]= 4 - 3 x^2 + x^3
```

```
In[3]:= Factor[f[x]]
|факторизовать
```

```
Out[3]= (-2 + x)^2 (1 + x)
```

Вычислим значения функции в заданных точках. Обратите внимание, как **Mathematica** выполняет подстановку иррациональных значений.

```
In[4]:= f[1.8]
```

```
Out[4]= 0.112
```

```
In[5]:= f[-E]
```

основание натурального логарифма

```
Out[5]= (1 - e) (4 + 4 e + e^2)
```

```
In[6]:= f[-E] // N
```

численное приближение

```
Out[6]= -38.2527
```

```
In[7]:= f[ $\sqrt[3]{14}$ ]
```

```
Out[7]= (1 + 141/3) (4 - 4 × 141/3 + 142/3)
```

```
In[8]:= f[ $\sqrt[3]{14}$ ] // N
```

численное приближение

```
Out[8]= 0.573643
```

Очистим переменные a и b , найдем $f(2a - b)$, упростим и раскроем скобки в полученном выражении.

```
In[9]:= Clear[a, b]; f[2 a - b]
```

очистить

```
Out[9]= (4 - 4 (2 a - b) + (2 a - b)^2) (1 + 2 a - b)
```

```
In[10]:= Simplify[f[2 a - b]]
```

упростить

```
Out[10]= (1 + 2 a - b) (2 - 2 a + b)^2
```

```
In[11]:= Expand[f[2 a - b]]
```

раскрыть скобки

```
Out[11]= 4 - 12 a^2 + 8 a^3 + 12 a b - 12 a^2 b - 3 b^2 + 6 a b^2 - b^3
```

Составим с помощью функции **Table** таблицу, содержащую координаты точек графика функции $f(x)$ на заданном отрезке с шагом 1 (используется по умолчанию). Выведем результат в табличной форме.

```
In[12]:= t1 = Table[{x, f[x]}, {x, -3, 4}]
```

таблица значений

```
Out[12]= {{-3, -50}, {-2, -16}, {-1, 0}, {0, 4}, {1, 2}, {2, 0}, {3, 4}, {4, 20}}
```

```
In[13]:= t1 // TableForm
```

табличная форма

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

-3	-50
-2	-16
-1	0
0	4
1	2
2	0
3	4
4	20

Составим таблицу значений функции $f(x)$ с шагом 0,5 по x .

```
In[14]:= t2 = Table[{x, f[x]}, {x, -3, 4, 0.5}]
```

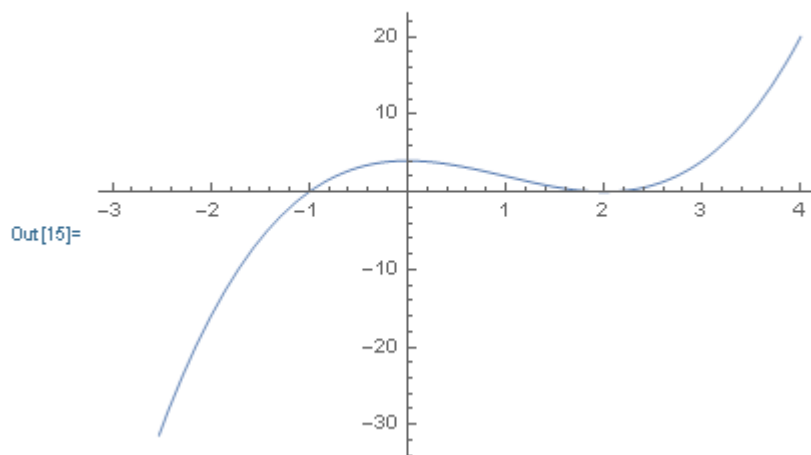
таблица значений

```
Out[14]:= {{-3., -50.}, {-2.5, -30.375}, {-2., -16.},
           {-1.5, -6.125}, {-1., 0.}, {-0.5, 3.125}, {0., 4.},
           {0.5, 3.375}, {1., 2.}, {1.5, 0.625}, {2., 0.},
           {2.5, 0.875}, {3., 4.}, {3.5, 10.125}, {4., 20.}}
```

Построим график функции $f(x)$ на заданном отрезке с помощью функции **Plot**.

```
In[15]:= gr1 = Plot[f[x], {x, -3, 4}]
```

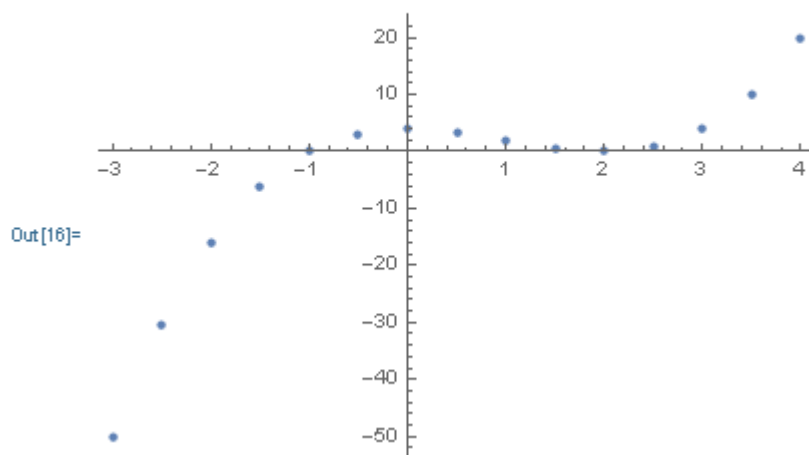
график функции



Применим функцию **ListPlot** для построения точек графика функции $f(x)$, соответствующие табличным значениям.

```
In[16]:= gr2 = ListPlot[t2]
```

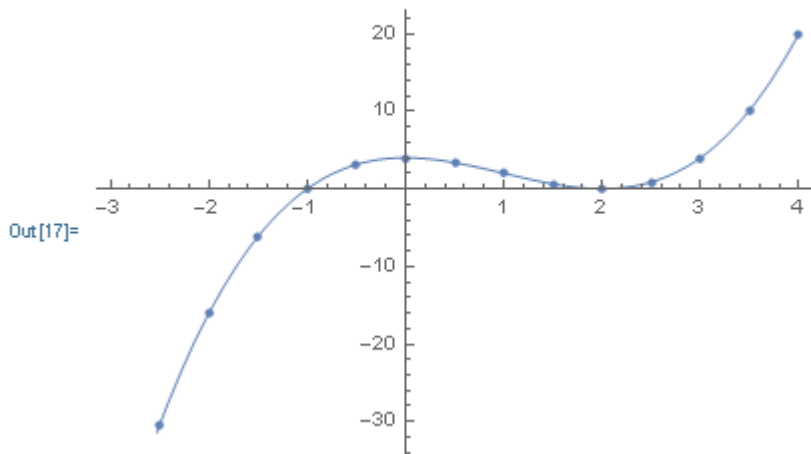
диаграмма разброса



Объединим оба рисунка в одном графическом окне с помощью функции **Show**.

In[17]:= Show[gr1, gr2]

показать



Пример 2.3. Задать функцию двух переменных $g(x, y) = 2x^2 + 3xy - 5y^2 + 7$. Найти значения функции в точках $(-2, 7; -1, 4)$, $(2/3; \ln 5)$, $(\sqrt{17}, \sin 2)$. Составить таблицу из координат точек графика функции $g(x, y)$ в прямоугольнике $[-3, 3] \times [-4, 4]$ с шагом 1 по x и шагом 2 по y . Построить график функции $g(x, y)$ в прямоугольнике $[-10, 10] \times [-15, 15]$. Построить линии уровня (контурный график) функции $g(x, y)$ в прямоугольнике $[-5, 5] \times [-5, 5]$. Построить кривую $g(x, y) = 0$ в прямоугольнике $[-8, 8] \times [-8, 8]$.

Δ Очистим все переменные и зададим функцию $g(x, y)$.

In[1]:= ClearAll;

очистить всё

In[2]:= g[x_, y_] = 2 x^2 + 3 x y - 5 y^2 + 7

Out[2]= 7 + 2 x^2 + 3 x y - 5 y^2

Обратите внимание, что произведение переменных x и y должно записываться с помощью пробела между ними либо знака умножения в виде точки. В противном случае запись «xy» **Mathematica** воспринимает как одну новую переменную.

Вычислим значения функции $g(x, y)$ в заданных точках.

In[3]:= g[-2.7, -1.4]

Out[3]= 23.12

In[4]:= g[2/3, Log[5]]

натуральн

In[5]:= g[2/3, Log[5]] // N

натуральн... численное приближение

Out[4]= $\frac{71}{9} + 2 \text{Log}[5] - 5 \text{Log}[5]^2$

Out[5]= -1.84369

```
In[6]:= g[ $\sqrt{17}$ , Sin[2]]
```

синус

```
In[7]:= g[ $\sqrt{17}$ , Sin[2]] // N
```

синус

численное приближение

```
Out[6]= 41 + 3  $\sqrt{17}$  Sin[2] - 5 Sin[2]2
```

```
Out[7]= 48.1133
```

Составим таблицу значений функции в заданном прямоугольнике.

```
In[8]:= tab = Table[{x, y, g[x, y]}, {x, -3, 3}, {y, -4, 4, 2}]
```

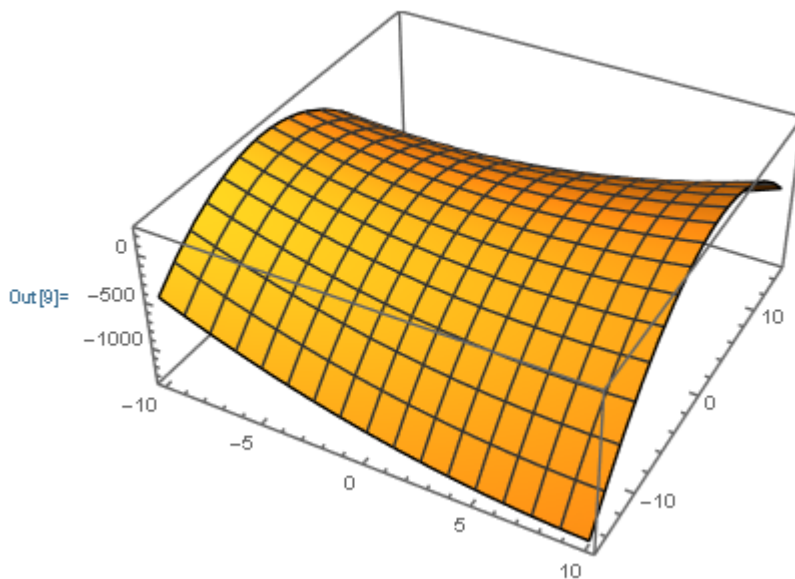
таблица значений

```
Out[8]= {{{-3, -4, -19}, {-3, -2, 23}, {-3, 0, 25}, {-3, 2, -13}, {-3, 4, -91}},
{{-2, -4, -41}, {-2, -2, 7}, {-2, 0, 15}, {-2, 2, -17}, {-2, 4, -89}},
{{-1, -4, -59}, {-1, -2, -5}, {-1, 0, 9}, {-1, 2, -17}, {-1, 4, -83}},
{{0, -4, -73}, {0, -2, -13}, {0, 0, 7}, {0, 2, -13}, {0, 4, -73}},
{{1, -4, -83}, {1, -2, -17}, {1, 0, 9}, {1, 2, -5}, {1, 4, -59}},
{{2, -4, -89}, {2, -2, -17}, {2, 0, 15}, {2, 2, 7}, {2, 4, -41}},
{{3, -4, -91}, {3, -2, -13}, {3, 0, 25}, {3, 2, 23}, {3, 4, -19}}}
```

Построим график функции $g(x, y)$ на заданном множестве (графиком функции двух переменных является поверхность в пространстве).

```
In[9]:= Plot3D[g[x, y], {x, -10, 10}, {y, -15, 15}]
```

график функции 2-х переменных



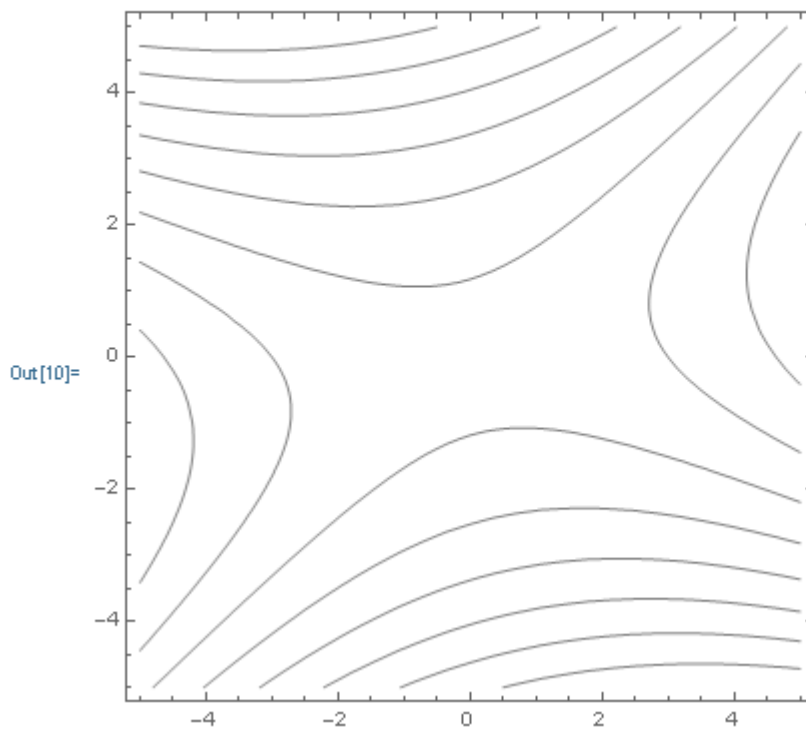
Графиком данной функции является гиперболический параболоид («седло»).

Построим линии уровня функции $g(x, y)$, т.е. множество кривых на плоскости OXY , определяемых уравнениями $g(x, y) = \text{const}$ (для всевозможных значений постоянной const). Или, другими словами, сечения графика функции плоскостями, параллельными плоскости OXY . Применим функцию **ContourPlot**.

```
In[10]:= ContourPlot[g[x, y], {x, -5, 5}, {y, -5, 5}, ContourShading -> False]
```

контурный график

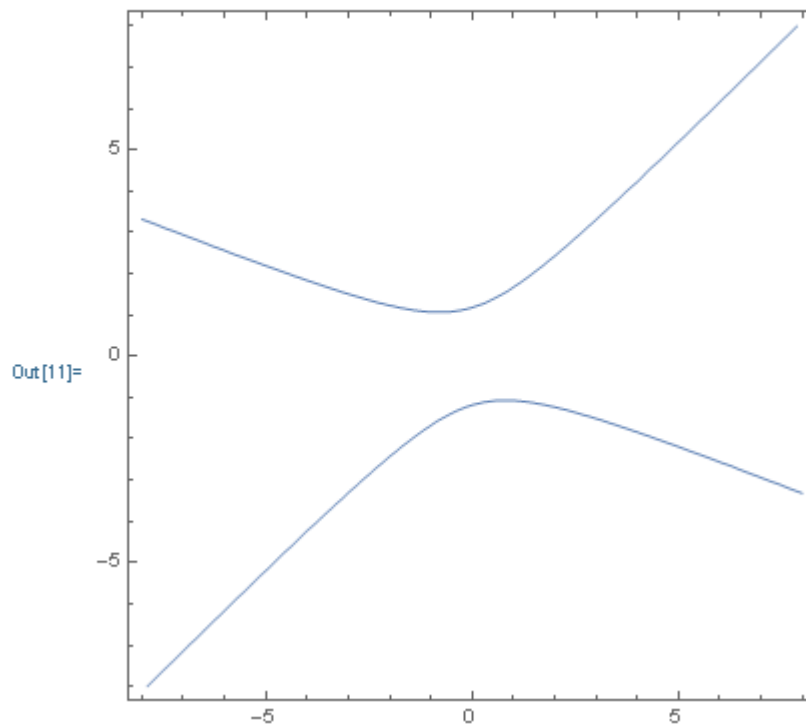
оцвечивание конт... [ложь]



Кривую $g(x, y) = 0$ будем рассматривать как линию нулевого уровня функции $g(x, y)$ и построим ее с помощью функции **ContourPlot**.

```
In[11]:= ContourPlot[g[x, y] == 0, {x, -8, 8}, {y, -8, 8}]
```

контурный график



Полученная кривая является гиперболой.

