Министерство образования и науки Украины
Харьковский национальный университет имени В. Н. Каразина

С. Ю. Игнатович
Р. Б. Райхцаум

Применение системы MAPLE для решения задач оптимизации

Методические указания
для студентов 4 курса механико-математического факультета
(специальность «Прикладная математика»)

Харьков – 2008

УДК 517 : 519.67 (076.5)
ББК 22.161 + 32.973.26 – 018.2
И 26

Утверждено на заседании Ученого совета механико-математического факультета Харьковского национального университета имени В. Н. Каразина (протокол № 6 от 20.06.2008)

Рецензенты: Ткачук Н. В., доктор технических наук, профессор кафедры АСУ Национального технического университета «ХПИ»;

Сохин А. С., кандидат физико-математических наук, доцент кафедры дифференциальных уравнений и управления Харьковского национального университета имени В. Н. Каразина.

И 26 **Игнатович С. Ю., Райхцаум Р. Б. Применение системы MAPLE для решения задач оптимизации:** Методические указания для студентов 4 курса механико-математического факультета (специальность «Прикладная математика»). – Х.: ХНУ имени В. Н. Каразина, 2008. – 48 с.

Цель данной разработки – помочь студентам в освоении математической системы символьных вычислений MAPLE.

На примере задачи нахождения экстремумов функций одной и двух переменных рассмотрены некоторые вычислительные и графические средства системы MAPLE. Предложены варианты заданий для самостоятельной работы.

УДК 517 : 519.67 (076.5)
ББК 22.161 + 32.973.26 – 018.2

Содержание

Введение

Изучение новых систем программирования можно сравнить с изучением иностранных языков: первый язык учить трудно, но чем больше языков знаешь, тем легче выучить еще один. Как и при изучении иностранного языка, при знакомстве с системой программирования хорошо иметь учебники и словари, однако полезно бывает просто «окунуться» в языковую среду, начать говорить на конкретные темы, зная всего несколько правил и ограниченное количество слов. Обычно самым трудным этапом освоения новой программы (как и нового иностранного языка!) является первый шаг, когда непонятно, с чего начать. Тут может прийти на помощь «разговорник», содержащий первые, самые полезные и самые часто используемые фразы.

Данное пособие и является таким «разговорником», который может быть использован на первых шагах изучения математической системы MAPLE. В нем описано применение MAPLE для решения задачи нахождения экстремумов функций одной и двух переменных. Как известно, MAPLE является системой компьютерной алгебры, она предназначена прежде всего для аналитических, символьных преобразований и вычислений. Важное преимущество MAPLE – мощные графические средства, которые позволяют получить наглядное представление о характере исследуемых функций, визуально локализовать решения. Мы предлагаем применить известные читателю аналитические методы при решении конкретных задач оптимизации с использованием этих средств. В системе MAPLE можно реализовать и численные методы. При этом уменьшается объем скучной «черновой» работы (в том числе для подготовки вывода графиков), которая так удручает пользователей при применении традиционных систем программирования.

На настоящий момент разработано несколько реализаций системы MAPLE (весной 2008 года вышла версия MAPLE 12), причем не все они полностью совместимы. Вид интерфейса, набор команд, внутренняя реализация алгоритмов в них различна, поэтому результаты выполнения конкретных команд для разных версий MAPLE могут быть различными. Мы настоятельно рекомендуем пользоваться встроенной справочной системой MAPLE (Help), которая, кроме собственно описания форматов команд, содержит большое количество примеров. Кроме того, написан ряд хороших учебников по MAPLE.

Некоторые из них перечислены в списке литературы.

Параграфы 1–6 настоящего пособия посвящены применению некоторых вычислительных и графических средств системы MAPLE для нахождения экстремумов функций одной и двух переменных. В параграфе 7 приведены задания для самостоятельного решения.

1. Функции одной переменной

Важным вопросом в исследовании функций является нахождение их максимумов и минимумов. Для гладких функций основным инструментом аналитического нахождения экстремумов является теорема Ферма.

Теорема (Ферма; необходимое условие экстремума первого порядка). *Пусть функция $f:\mathbb{R}\to\mathbb{R}$ в точке x_0 имеет локальный экстремум. Если в этой точке существует производная $f'(x_0)$, то $f'(x_0)=0$.*

Если в точке x_0 производная функции f равна нулю, точку называют критической или стационарной.

Для дважды дифференцируемых функций можно применить теорему о необходимых и достаточных условиях экстремума второго порядка.

Теорема (условия экстремума второго порядка). *Пусть функция $f:\mathbb{R}\to\mathbb{R}$ является дифференцируемой в окрестности точки x_0 и пусть существует $f(x_0)$.*

1. Если в точке x_0 функция f имеет локальный минимум, то $f'(x_0)=0$ и $f''(x_0)\geq 0$. Если в точке x_0 она имеет локальный максимум, то $f'(x_0)=0$ и $f''(x_0)\leq 0$.

2. Если $f'(x_0)=0$ и $f''(x_0)>0$, то функция f имеет в точке x_0 локальный минимум. Если $f'(x_0)=0$ и $f''(x_0)<0$, то функция f имеет в точке x_0 локальный максимум.

Рассмотрим пример нахождения экстремумов функции с использованием системы MAPLE. Объектом исследования будет функция $f(x)=\frac{1}{4}x^4-\frac{2}{3}x^3-\frac{3}{2}x^2+2$. Задача ставится следующим образом: найти точки экстремума функции $f(x)$ и ее экстремальные значения.

Зададим нашу функцию:

```
> restart;
> f:=1/4*x^4-2/3*x^3-3/2*x^2+2;
```

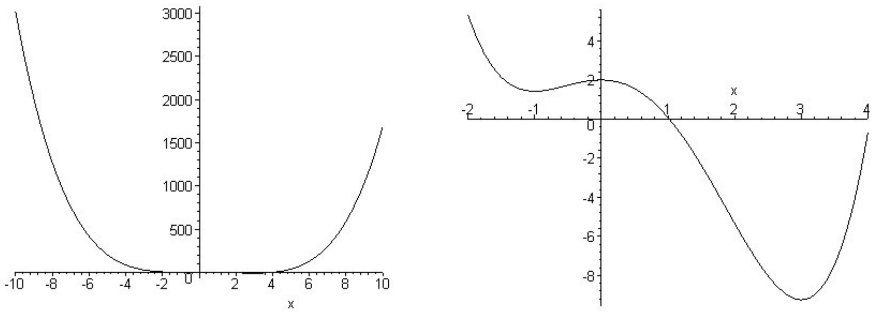


Рис. 1. Графики функции $f(x)$ на отрезках $[-10, 10]$ и $[-2, 4]$

и нарисуем ее график. Вначале нарисуем график функции на каком-нибудь промежутке, используя команду `plot`:

```
> plot(f,x=-10..10);
```

Получим график, изображенный на рис. 1 слева. Рассмотрим более подробно «подозрительный» промежуток:

```
> plot(f,x=-2..4);
```

Получим график, изображенный на рис. 1 справа.

Мы видим, что на данном промежутке функция имеет два локальных минимума и один локальный максимум. Из вида нашей функции ($f(x)$ – полином 4-й степени) следует, что других экстремумов у нее нет. Далее можно было бы локализовать каждый из них и найти его приближенно с помощью такого «визуального» метода.

Прежде чем локализовать критические точки, можно нарисовать график функции *на всей оси*:

```
> plot(f,x=-infinity..infinity);
```

По этой команде MAPLE попытается изобразить качественный вид графика функции. Это иногда может помочь определить, сколько экстремумов имеет данная функция[1].

[1] При этом масштабы по осям выбираются нестандартно: попробуйте, например, нарисовать график функции $\sin x$ на всей оси.

Давайте теперь применим аналитические методы для нахождения экстремумов нашей функции. Вначале воспользуемся теоремой Ферма. Для этого найдем производную функции $f(x)$ с помощью команды `diff`:

```
> df:=diff(f,x);
```

Получим

$$df := x^3 - 2x^2 - 3x.$$

Теперь найдем корни производной, то есть решим уравнение $f'(x) = 0$. Применим команду `solve`:

```
> solve(df=0);
```

Можно объединить эти два действия в одно:

```
> solve(diff(f,x)=0);
```

Получаем три корня: $0, 3, -1$. Это критические точки функции $f(x)$. Мы уже знаем (из вида графика), что -1 и 3 – это локальные минимумы, а 0 – это локальный максимум. Теперь, чтобы найти экстремальные значения функции, нужно критические точки по очереди подставить в функцию:

```
> x:=0: print('x'=x, 'f'=f);
> x:=3: print('x'=x, 'f'=f);
> x:=-1: print('x'=x, 'f'=f);
```

Обратите внимание на то, что после присвоения значений переменной x стоит двоеточие вместо точки с запятой. Это позволяет не выводить на печать промежуточные результаты. В данном случае выводом на печать управляет команда `print`. Конечно, это сделано только для удобства. Те же результаты, но в несколько менее удобной форме мы получим, выполняя такие команды:

```
> x:=0; f;
> x:=3; f;
> x:=-1; f;
```

Итак, имеем критические точки и соответствующие значения функции:

$$x = 0,\ f = 2,$$
$$x = 3,\ f = -\tfrac{37}{4},$$
$$x = -1,\ f = \tfrac{17}{12}.$$

Проверим условия экстремума второго порядка. Для нахождения второй производной используем команду `diff(f,x$2)`, в которой символ `$2` означает, что ищется производная второго порядка. Найдем знак второй производной (с помощью команды `sign`) в каждой из критических точек:

```
> x:='x';
> ddf:=diff(f,x$2);
> x:=0: print('x'=x,'sign(ddf)'=sign(ddf));
> x:=3: print('x'=x,'sign(ddf)'=sign(ddf));
> x:=-1: print('x'=x,'sign(ddf)'=sign(ddf));
```

Обратите внимание на то, что перед применением команды `diff` используется операция `x:='x'`, которая *восстанавливает x как переменную* после того, как ранее в программе ей было присвоено конкретное значение. Получим:

$$x = 0,\ sign(ddf) = -1,$$
$$x = 3,\ sign(ddf) = 1,$$
$$x = -1,\ sign(ddf) = 1.$$

Итак, при $x = 0$ функция имеет локальный максимум, а при $x = 3$ и $x = -1$ – локальный минимум.

Полный текст программы MAPLE см. в приложении, стр. 43.

Замечание. В этом параграфе мы рассмотрели *один простой* пример нахождения экстремумов функции одной переменной. Поскольку наша функция является полиномом, мы заранее знали, что она имеет конечное количество экстремумов (в нашем случае – не более трех). Более того, MAPLE очень хорошо справляется с нахождением *всех* корней *полинома*, а значит, аналитический метод нахождения экстремума достаточно эффективен. Если же исследуемая функция является более сложной, придется обратить особое внимание на нахождение корней производной. Рассмотрим, например, функцию $f(x) = \sin(x^2)$ и найдем корни ее производной:

```
> f:=sin(x^2);
> df:=diff(f,x); solve(df=0);
```

Получаем три корня: 0, $\frac{1}{2}\sqrt{2}\sqrt{\pi}$, $-\frac{1}{2}\sqrt{2}\sqrt{\pi}$. Из вида графика ясно, что 0 – точка (локального) минимума, а $\pm\frac{1}{2}\sqrt{2}\sqrt{\pi}$ – точки максимума. Однако нетрудно видеть, что наша функция имеет бесконечное число экстремумов. Чтобы их найти, можно применить функцию `fsolve`, в которой указать отрезок, на котором ищется корень. Например, по команде

```
> fsolve(df=0,x=-2.5..-2);
```

найдем еще один корень производной, а именно, -2.170803764. Заметим, что команда `fsolve` выдает приближенное решение (в отличие от команды `solve`, по которой может быть получено и аналитическое решение, как для функции $f(x) = \sin(x^2)$). Более того, команда `fsolve` выдает только один корень функции на указанном интервале, а иногда не может его найти, даже если он там есть! Поэтому в случае достаточно сложной функции при нахождении экстремумов особое внимание приходится уделять корректному применению команд `solve` и `fsolve`.

Замечание. Если корень функции локализован, то для его нахождения можно применять и приближенные методы, используя систему MAPLE. Вот, например, простейшая программа, реализующая метод половинного деления для нахождения корня функции df, если уже известно, что он находится на отрезке $[1.5, 4]$, причем $df(1.5) < 0$ и $df(4) > 0$:

```
> a:=1.5; b:=4;
> for i from 1 to 30 do x:=(a+b)/2:
if (df>0) then b:=(a+b)/2 else a:=(a+b)/2 fi od:
> (a+b)/2;
```

Данная конструкция цикла означает, что переменная i пробегает все значения от 1 до 30, т. е. тело цикла выполняется 30 раз.

Команды `do` и `od` – это *операторные скобки*, отмечающие начало и конец тела цикла (вместо команды `od` можно использовать более длинную команду `end do`). Продемонстрируем другой вариант использования оператора цикла:

```
> while b-a>0.00001 do x:=(a+b)/2:
if (df>0) then b:=(a+b)/2 else a:=(a+b)/2 fi od:
```

Кстати, для более детального *численного* анализа (в том числе при применении приближенных методов) может пригодиться переменная `Digits`, которая устанавливает количество выводимых цифр числа. По умолчанию этой переменной присваивается значение 10. Чтобы выводить больше цифр, нужно (лучше в начале программы) присвоить переменной `Digits` большее значение.

2. Функции нескольких переменных

Напомним необходимые и достаточные условия локального экстремума для функций нескольких переменных.

Теорема (необходимое условие экстремума первого порядка). *Пусть функция $f : \mathbb{R}^n \to \mathbb{R}$ в точке x_0 имеет локальный экстремум. Если в этой точке существуют частные производные $\frac{\partial f(x_0)}{\partial x_i}$, $i = 1, \dots, n$, то все эти производные равны нулю.*

Если в точке x_0 все частные производные функции f равны нулю, точку называют критической или стационарной.

Теорема (условия экстремума второго порядка). *Пусть функция $f : \mathbb{R}^n \to \mathbb{R}$ определена, непрерывна и имеет непрерывные частные производные первого и второго порядка в некоторой окрестности точки x_0.*

1. Если в точке x_0 она имеет локальный минимум, то

$$\frac{\partial f(x_0)}{\partial x_i} = 0, \quad i = 1, \dots, n, \tag{$*$}$$

и матрица частных производных второго порядка

$$\begin{pmatrix} \frac{\partial^2 f(x_0)}{\partial x_1^2} & \frac{\partial^2 f(x_0)}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f(x_0)}{\partial x_1 \partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 f(x_0)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x_0)}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f(x_0)}{\partial x_n^2} \end{pmatrix} \tag{$**$}$$

неотрицательно определена.

Если в точке x_0 она имеет локальный максимум, то выполнены условия $(*)$ и матрица $(**)$ неположительно определена.

2. Если выполнены условия $(*)$ и матрица $(**)$ положительно определена, то функция f имеет в точке x_0 локальный минимум. Если выполнены условия $(*)$ и матрица $(**)$ отрицательно определена, то функция f имеет в точке x_0 локальный максимум.

Мы ограничимся рассмотрением функций двух переменных. Приведем пример исследования функции двух переменных с использованием системы MAPLE.

Зададим функцию. Это можно сделать так же, как на стр. 5:

```
> f:=x^4+y^4-2*x^2-2*y^2+4*x*y;
```

Как и ранее, для получения значения функции f в конкретной точке (например, (1,1)) нужно выполнить такие команды:

```
> x:=1; y:=1; f;
```

Однако в этом параграфе мы продемонстрируем использование другого способа задания функции – с помощью *функционального оператора*:

```
> f:=(x,y)->x^4+y^4-2*x^2-2*y^2+4*x*y;
```

При обращении к функции, заданной таким образом, нужно в скобках указывать список ее аргументов. Например, по команде

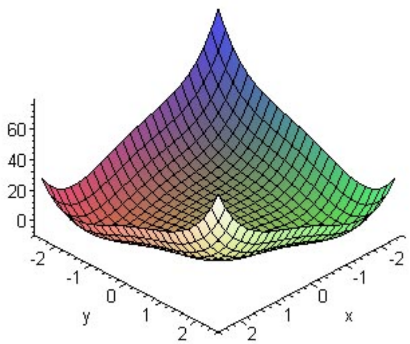
```
> f(1,1);
```

мы получим значение функции f в точке $x = 1$, $y = 1$.

Прежде всего нарисуем (трехмерный) график нашей функции:

```
> plot3d(f(x,y),x=-2.5..2.5,y=-2.5..2.5,axes=frame);
```

В этой команде указана опция `axes`, которая задает вид координатных осей. На самом деле, в команде `plot3d`, как и во многих других графических командах, можно указывать большое количество различных опций, которые задают вид графиков: цвет, тип и толщину линий, угол поворота осей и др. Их подробное описание можно найти в Help.



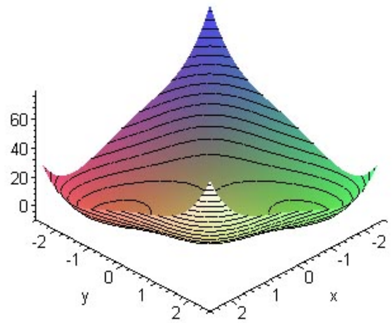


Рис. 2. Трехмерный график функции $f(x,y)$

Итак, мы получаем график, изображенный на рис. 2 слева. Заметим, что MAPLE позволяет поворачивать готовый график с помощью мыши, чтобы рассмотреть его с разных сторон.

На этой поверхности для наглядности нанесена координатная сетка, т. е. нарисованы линии $x = const$ и $y = const$. Другой способ визуализации поверхности – нанести на поверхность линии $z = const$ (они называются *линиями уровня* функции f). Получившийся график называется *контурным графиком*. Контурный график можно построить с помощью команды `contourplot3d`, которая на поверхности рисует линии уровня, получающиеся как пересечение поверхности и семейства *равноотстоящих* плоскостей, параллельных плоскости xOy. Для того, чтобы использовать это средство MAPLE, нужно вначале подключить пакет расширения `plots` с помощью команды `with`:

```
> with(plots):
> contourplot3d(f(x,y),x=-2.5..2.5,y=-2.5..2.5,
axes=frame,filled=true,contours=45);
```

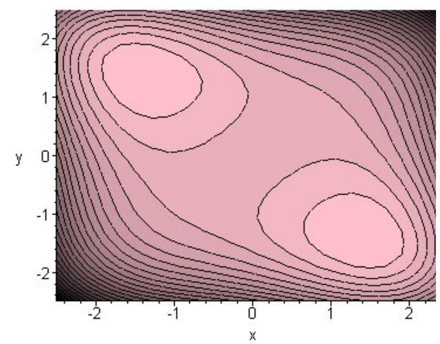
Опция `contours` задает количество линий уровня функции на поверхности.

Другой способ получить контурный график – воспользоваться командой plot3d, в которой опции style присвоено значение patchcontour:

```
> plot3d(f(x,y),x=-2.5..2.5,y=-2.5..2.5,axes=frame,
style=patchcontour,contours=45);
```

По этой команде получим график, изображенный на рис. 2 справа. На контурном графике чуть лучше виден характер поведения функции, но все равно для определения экстремумов удобнее использовать двумерные линии уровня функции. Их можно получить таким образом:

```
>contourplot(f(x,y),x=-2.5..2.5,y=-2.5..2.5,axes=frame,
filled=true,contours=25,coloring=[yellow,green]);
```



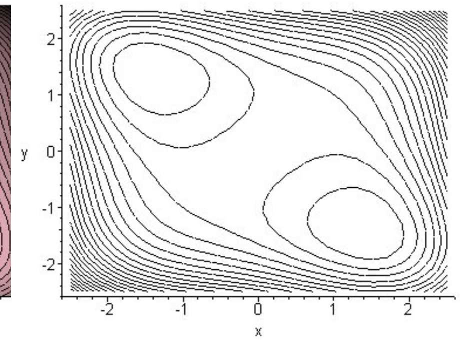


Рис. 3. Двумерные графики линий уровня функции $f(x,y)$

Получим картинку, изображенную на рис. 3 слева. В этой команде опция filled равна true. Это означает, что MAPLE автоматически закрашивает области между линиями уровня в разные цвета в зависимости от значения функции f на этих линиях уровня. Это делает график более выразительным и позволяет визуально оценить характер изменения значений функции. Если опцию filled положить равной false (или вообще не указывать), то получим «незакрашенный» график (как на рис. 3 справа). Напротив, если положить filled=true, то можно использовать опцию coloring, которая позволяет управлять цветами закраски. Например, если указана `coloring=[yellow,green]`, то линии уровня, отвечающие наименьшему значению f, будут желтыми, линии уровня, отвечающие наибольшему значению f – зелеными, а все промежуточные линии уровня MAPLE закрасит в промежуточные цвета.

Глядя на эти рисунки, можно высказать предположение, что наша функция имеет два экстремума (они лежат внутри замкнутых кривых), причем из характера графика, полученного выше, следует, что это – точки минимума. Теперь можно, варьируя пределы изменения x и y, рассмотреть подробнее «подозрительные участки», или наоборот, построить линии уровня функции вне области, изображенной выше, чтобы убедиться в том, что там функция экстремумов не имеет.

Далее продемонстрируем, как можно решать задачу нахождения экстремумов нашей функции аналитически. Вначале применим необходимое условие первого порядка. Для нахождения критических точек функции зададим систему уравнений $\frac{\partial f(x,y)}{\partial x} = 0$, $\frac{\partial f(x,y)}{\partial y} = 0$:

```
> Sys_Eqn:=diff(f(x,y),x)=0,diff(f(x,y),y)=0;
```

Отметим, что уравнения системы перечисляются после знака `:=` через запятую. Получаем:

$$Sys\_Eqn := 4x^3 - 4x + 4y = 0,\ 4y^3 - 4y + 4x = 0.$$

Для решения этой системы применим команду `solve`:

```
> solve({Sys_Eqn});
```

Заметим, что в этой команде уравнения системы заключены в фигурные скобки, которые означают тип данных «множество».

По команде `solve` MAPLE пытается найти точные решения системы, если это возможно. Однако в данном случае MAPLE выдаст сообщение, содержащее, в частности, функцию *RootOf*. Это означает, что аналитические решения системы нельзя выразить в радикалах. В таком случае можно попытаться получить численные, приближенные значения решений с помощью команды `evalf`. Эта команда служит для принудительного приближенного вычисления результата:

```
> evalf(%);
```

Обратим внимание на символ %, который стоит в качестве параметра. Он означает результат последнего **по времени** вычисления[2].

Две последние команды можно объединить в одну:

```
> evalf(solve({Sys_Eqn}));
```

Другой способ «бороться» с функцией *RootOf* в ответе – в исходную систему искусственно внести «неточность», чтобы MAPLE не пыталась найти точное решение. Это можно сделать, например, так: вместо какого-нибудь целого числа в записи уравнения системы, например, числа 0, написать число 0.0 (или просто 0. – «ноль с точкой»). MAPLE воспринимает это число как нецелое, и автоматически ищет приближенные решения такой системы.

Итак, мы получили такие (вещественные) корни: точку $(0, 0)$ (кратности 3) и точку $(-1.414213562, 1.414213562)$. Остается открытым вопрос: получены ли *все* решения системы? Решение систем нелинейных уравнений – очень сложная задача, и универсальные методы, которые применяются в системе MAPLE, могут выдать только часть решений. Для контроля применим графический метод: на плоскости (x, y) нарисуем кривые, отвечающие уравнениям нашей системы. Воспользуемся командой построения графиков неявных функций `implicitplot` (эта команда также находится в пакете `plots`):

```
> implicitplot({Sys_Eqn},x=-2.5..2.5,y=-2.5..2.5);
```

[2] Заметим, что пользователь может выполнять команды не в той последовательности, в которой они введены и изображены на экране. Поэтому конкретное значение выражения % зависит от того, какая команда была выполнена последней, а не от того, что написано в предыдущей строке.

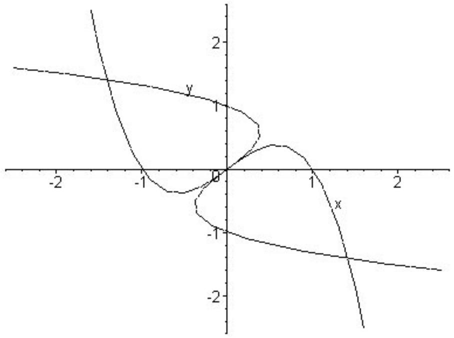


Рис. 4. Кривые $\frac{\partial f(x,y)}{\partial x} = 0$ и $\frac{\partial f(x,y)}{\partial y} = 0$

На рисунке мы видим две кривые, которые задаются уравнениями $4x^3 - 4x + 4y = 0$ и $4y^3 - 4y + 4x = 0$ нашей системы. Значит, решения нашей системы – это точки пересечения кривых. Итак, рассматриваемая система имеет по меньшей мере три решения: начало координат и еще две точки, симметричные относительно начала координат. Выше мы видели, что MAPLE по команде `evalf(%)` выдает только два из этих трех корней![3]

Когда корни локализованы, для нахождения их численных значений гораздо эффективнее вместо команды `solve` применять команду `fsolve`, в которой можно указать область для поиска решений:

```
> fsolve({Sys_Eqn},{x=0.1..2.5,y=-2.5..0.1});
```

Аналогично можно получить и остальные решения. Заметим, что команда `fsolve` не всегда находит решения в указанной области, даже если они там есть. В таких случаях помогает такой прием: попытаться изменить пределы варьирования переменных. Например, для нахождения решения $(0, 0)$ команда

```
> fsolve({Sys_Eqn},{x=-0.5..0.5,y=-0.5..0.5});
```

может не выдать решения[4] (тогда MAPLE в строке результата просто повторяет введенную команду), а по команде

[3] Это может зависеть от версии MAPLE.
[4] Это зависит от версии MAPLE.

```
> fsolve({Sys_Eqn},{x=-0.05..0.05,y=-0.05..0.05});
```

результат может быть успешно получен.

Итак, мы получили, что критическими точками нашей функции $f(x,y)$ являются точки

$$(0,0),\ (1.414213562, -1.414213562),\ (-1.414213562, 1.414213562).$$

Так как $\sqrt{2} \approx 1.414213562$, то возникает предположение, что на самом деле три решения системы – это точки

$$(0,0), \quad (\sqrt{2}, -\sqrt{2}), \quad (-\sqrt{2}, \sqrt{2}).$$

Чтобы это проверить, подставим указанные точки в систему, используя команду `subs`:

```
> subs(x=sqrt(2),y=-sqrt(2),Sys_Eqn);
```

Получим $0 = 0$, значит, действительно, $(\sqrt{2}, -\sqrt{2})$ – решение системы. Аналогично проверяется точка $(-\sqrt{2}, \sqrt{2})$.

И тем не менее, вопрос о том, найдены ли все решения системы `Sys_Eqn`, остается открытым! Для того, чтобы доказать, что мы нашли все решения системы, примем во внимание конкретный вид уравнений. В *данном конкретном* случае можно сложить уравнения системы, откуда сразу получим, что $x^3+y^3 = 0$, а значит, $x = -y$. Подставляя это выражение для x, например, в первое уравнение, получаем $-4y^3 + 8y = 0$, откуда $y = 0$, $y = \pm\sqrt{2}$. Итак, действительно, наша система имеет ровно три решения, которые мы уже нашли.

Далее применим условия экстремума второго порядка. Для этого построим матрицу вторых производных, используя тип данных `matrix`:

```
> My_matr:=matrix(2,2,
[[diff(f(x,y),[x,x]),diff(f(x,y),[x,y])],
[diff(f(x,y),[y,x]),diff(f(x,y),[y,y])]]);
```

Первые два параметра описания типа `matrix` (в данном случае они равны 2, 2) – это размеры матрицы (количество строк и столбцов), а третий параметр – это список строк.

Тип данных «список» в системе MAPLE задается таким образом: все элементы списка помещаются в квадратные скобки. Например, команда

```
> My_List:=[1,2,3];
```

задает список из трех элементов – чисел 1, 2 и 3, а команда

```
> My_List:=[[1,2,3],[4,5,6]];
```

задает список из двух элементов, причем каждый из этих элементов – это тоже список.

Значит, в описании матрицы `My_matr` третий параметр – это «список списков»: внешний список – это список строк, а каждая строка – это список элементов, входящих в эту строку. Обратите внимание на то, что в уже известной нам команде `diff` для нахождения производных второго порядка используется *список* переменных, по которым нужно производить дифференцирование.

Итак, мы определили переменную `My_matr`, которая имеет вид

$$My\_matr := \begin{bmatrix} 12x^2-4 & 4 \\ 4 & 12y^2-4 \end{bmatrix}.$$

Для того, чтобы определить, является ли матрица вторых производных положительно определенной или отрицательно определенной, воспользуемся критерием Сильвестра. Для этого найдем определитель нашей матрицы. Подключив пакет расширения `linalg`, используем функцию `det`:

```
> with(linalg):
> d:=det(My_matr);
```

Получим:

$$d := 144x^2y^2 - 48x^2 - 48y^2.$$

Теперь подставим по очереди все критические точки в главные угловые миноры нашей матрицы:

```
> x:=0:y:=0: print('x'=x,'y'=y,
'First_minor'=My_matr[1,1],'Second_minor'=d);
> x:=sqrt(2):y:=-sqrt(2): print('x'=x,'y'=y,
'First_minor'=My_matr[1,1],'Second_minor'=d);
> x:=-sqrt(2):y:=sqrt(2): print('x'=x,'y'=y,
'First_minor'=My_matr[1,1],'Second_minor'=d);
```

Это же можно сделать в цикле: зададим список `My_List` критических точек, в котором каждый элемент списка – список из двух координат точки, а потом выполним цикл по переменной `z` (она имеет тип «список»), которая пробегает список наших критических точек:

```
>My_List:=[[0,0],[sqrt(2),-sqrt(2)],[-sqrt(2),sqrt(2)]];
>for z in My_List do x:=z[1];y:=z[2];print('x'=x,'y'=y,
'First_minor'=My_matr[1,1],'Second_minor'=d) od:
```

Получаем, что матрицы вторых производных в точках $(\sqrt{2}, -\sqrt{2})$ и $(-\sqrt{2}, \sqrt{2})$ положительно определены. Следовательно, функция f имеет локальные минимумы в этих точках. Минимальные значения можно найти так:

```
> f(sqrt(2),-sqrt(2)); f(-sqrt(2),sqrt(2));
```

В точке $(0, 0)$ матрица вторых производных вырожденная. Значит, для того, чтобы определить, есть ли в этой точке локальный экстремум, необходимо провести дополнительные исследования. Покажем, как можно использовать MAPLE в такой ситуации.

Прежде всего, можно рассмотреть более подробно график функции f в окрестности точки $(0, 0)$, используя команды `plot3d`, `contourplot3d` и `contourplot`. Можно также попробовать нарисовать отдельно линию уровня функции, на которой лежит точка (0,0), т. е. кривую $f(x, y) = 0$. Это можно сделать с помощью команды `implicitplot`:

```
> implicitplot(f(x,y)=0,x=-3..3,y=-3..3,grid=[30,30]);
```

Эта команда работает следующим образом: вначале определяется знак $f(x,y)$ в точках сетки $n\times m$, где величины n и m определяются опцией `grid=[n,m]`. Если в соседних точках функция f имеет разные знаки, то между ними должна находиться точка кривой $f(x,y)=0$. По определенному интерполяционному алгоритму MAPLE строит *отдельные точки* искомой кривой, а затем эти точки соединяются так, чтобы получилась гладкая кривая. Для кривых с особенностями эта команда может работать не совсем корректно, особенно в случае разрывных кривых и кривых с самопересечениями. Тогда можно попробовать увеличить число точек сетки (с помощью опции `grid`), однако при этом время выполнения команды заметно возрастет. Кроме того, можно использовать опцию `style=point`. Тогда MAPLE не будет соединять точки, а нарисует на их месте кружочки. В некоторых случаях такой рисунок будет более наглядным.

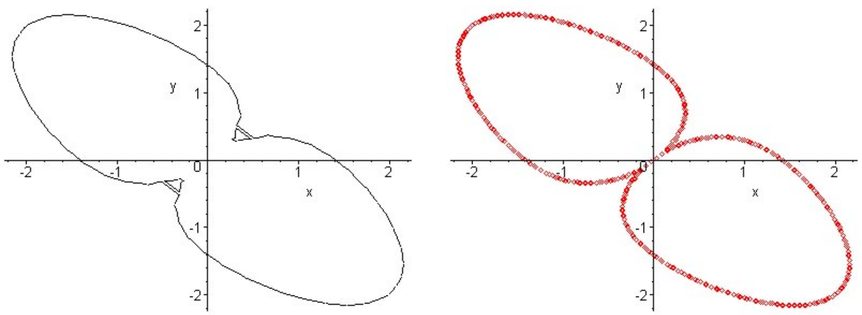


Рис. 5. Линия уровня $f(x,y)=0$

На рис. 5 показаны результаты выполнения команды `implicitplot`: график слева соответствует команде

```
> implicitplot(f(x,y)=0,x=-3..3,y=-3..3,grid=[30,30]);
```

а график справа – команде

```
> implicitplot(f(x,y)=0,x=-3..3,y=-3..3,grid=[100,100],
style=point);
```

Итак, линия уровня $f(x, y) = 0$ состоит из двух замкнутых кривых, имеющих общую точку (0,0). Так может выглядеть линия уровня, соответствующая седлу: в некоторых направлениях функция f растет, а в некоторых – убывает, т. е. (0,0) не является точкой экстремума. Докажем, что это действительно так.

Вернемся еще раз к рис. 3, на котором изображены линии уровня функции $f(x, y)$, и попробуем найти направления роста и убывания функции (из начала координат). Глядя на график, можно предположить, что вдоль прямых $y = x$ функция растет, а вдоль прямых $y = -x$ – убывает. Покажем это. Определим функции

```
> f1:=(x)->f(x,x);
> f2:=(x)->f(x,-x);
```

и нарисуем их графики в окрестности точки $x = 0$ (или исследуем эти функции так, как рассказано в параграфе 1). Получаем, что в точке $x = 0$ функция $f1$ имеет минимум, а функция $f2$ – максимум. Следовательно, точка (0,0) действительно не является экстремальной для исходной функции f.

Полный текст программы MAPLE см. в приложении, стр.43.

Замечание. В этом параграфе мы продемонстрировали возможности MAPLE для нахождения экстремумов *полиномиальной* функции двух переменных. В таком случае при применении метода множителей Лагранжа возникает система полиномиальных уравнений. Обычно MAPLE хорошо справляется с нахождением решений полиномиальных систем. Однако в более сложных случаях особое внимание приходится уделять локализации решений систем, чтобы в дальнейшем можно было корректно применять команду `fsolve` в ограниченных областях. Здесь особенно полезными оказываются возможности MAPLE по визуализации поверхностей.

Интересный класс поверхностей можно получить таким образом. Рассмотрим функцию h от двух переменных x и y, зависящую еще от трех параметров:

```
> h:=(x,y,a,b,c)->c*exp(-(x-a)^2-(y-b)^2);
```

и зададим функцию $f(x,y)$ как сумму нескольких таких слагаемых, отвечающих различным значениям параметров, например:

```
>f:=(x,y)->h(x,y,-1,-1,1)+h(x,y,0,0,-1.2)+h(x,y,2,-2,1);
```

На рис. 6 изображены ее график и линии уровня. Полученная «вулканическая» поверхность похожа на фантастический пейзаж.

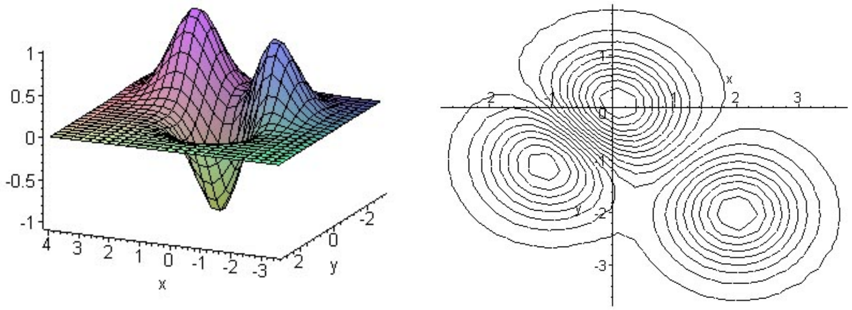


Рис. 6. «Вулканическая» поверхность

Замечание. Если точка экстремума локализована, то для ее нахождения можно применять численные методы, используя систему MAPLE. В качестве примера приведем одну возможную реализацию градиентного метода для нахождения минимума нашей функции f. Напомним, что идея градиентного метода состоит в следующем: чтобы найти очередное приближение, мы «шагаем» из текущей точки в направлении антиградиента функции f. При этом величина шага может выбираться поразному. В нашей программе она выбирается так, чтобы значение функции в каждом следующем приближении было меньше, чем в предыдущем:

```
> restart;
> f:=(x,y)->x^4+y^4-2*x^2-2*y^2+4*x*y;
> fx:=diff(f(x,y),x); fy:=diff(f(x,y),y);
> x0:=2.0:y0:=-1.0:List[0]:=[x0,y0]:
> w:=1:j:=0:
```

```
> while w>0.00001 do x:=x0:y:=y0:a:=1:
for i from 1 to 20 do x1:=x0-a*fx:y1:=y0-a*fy:
if f(x1,y1)-f(x0,y0)<0 then break else a:=a/2:fi:od:
w:=sqrt((x1-x0)^2+(y1-y0)^2):x0:=x1:y0:=y1:j:=j+1:
List[j]:=[x0,y0]: od:
> j;x0;y0;
> with(plots):with(plottools):
> for k from 0 to j do
d[k]:=display(POINTS(List[k])):od:
> display(seq([d[k]],k=0..j),insequence=true);
> for k from 1 to j do
c[k]:=display(line(List[k-1],List[k])):od:
> x:='x':y:='y':
z:=contourplot(f(x,y),x=0.5..2,y=-1.6..-0.9,
contours=18,scaling=constrained):
> display(z,seq([c[k]],k=1..j));
```

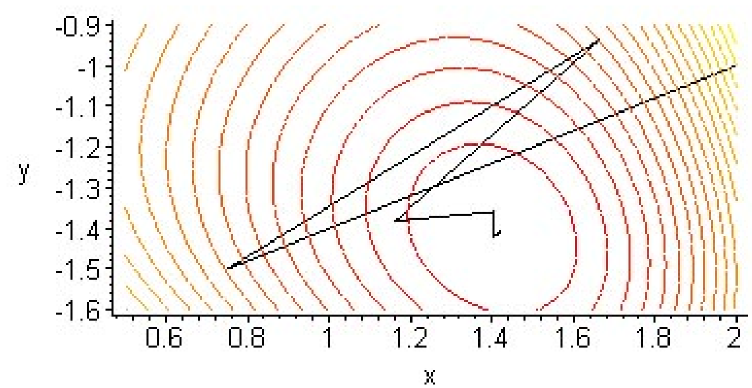


Рис. 7. Нахождение минимума градиентным методом

В этой программе (fx, fy) – это градиент функции, $(x0, y0)$ и $(x1, y1)$ – два последовательных приближения, j – номер итерации, w – точность, a – величина шага в направлении антиградиента. Список *List* создается для дальнейшей визуализации приближений. Команда

```
> display(seq([d[k]],k=0..j),insequence=true);
```

позволяет получить *анимацию* приближений: точки, приближающие решение, выводятся последовательно. Анимационный эффект достигается благодаря значению `true` опции `insequence` (по умолчанию эта опция имеет значение `false`). Команда

```
> display(z,seq([c[k]],k=1..j));
```

рисует одновременно все последовательные приближения, соединенные отрезками, на фоне линий уровня (см. рис. 7).

3. Пример: функция Розенброка

Как и в случае функции одной переменной, для полиномов от нескольких переменных аналитический метод нахождения экстремума *обычно* оказывается достаточно эффективным. Однако при визуальном способе локализации корней даже в этом случае могут возникнуть трудности. Классический пример – функция Розенброка:

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2.$$

Эта функция имеет *овражный характер*: ее график напоминает пологий изогнутый овраг с почти плоским дном и крутыми стенками. Функция Розенброка используется как тестовая для проверки и сравнения эффективности различных *численных* методов нахождения экстремумов. Из вида функции Розенброка следует, что она имеет минимум в точке $x = 1$, $y = 1$, однако простого взгляда на график недостаточно, чтобы этот минимум локализовать.

Посмотрим, можно ли «увидеть» овраги функции Розенброка с помощью системы MAPLE. Сначала нарисуем ее график:

```
> f:=(x,y)->100*(y-x^2)^2+(1-x)^2;
> plot3d(f(x,y),x=-5..5,y=-12..12,axes=framed);
```

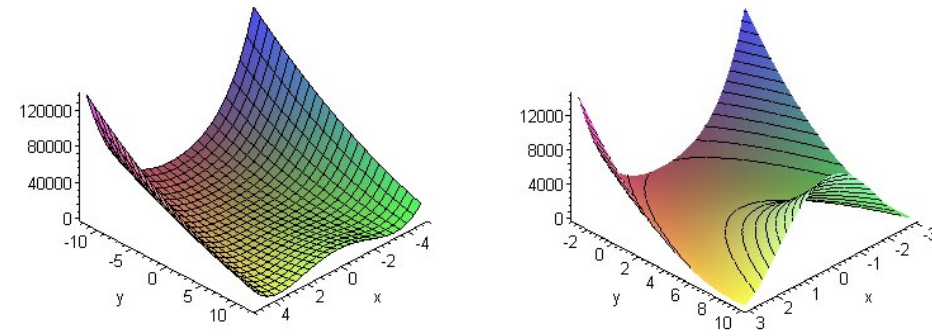


Рис. 8. График функции Розенброка

Полученный график (рис. 8 слева) показывает, что функция очень быстро возрастает, но визуально точку минимума определить не удается. Уменьшим пределы изменения аргументов и применим опцию `style=patchcontour`:

```
> plot3d(f(x,y),x=-3..3,y=-3..10,axes=framed,
style=patchcontour);
```

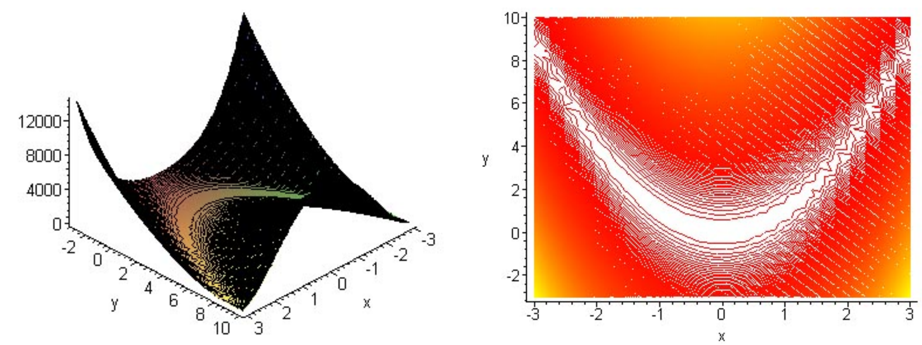


Рис. 9. Дно оврага (функция Розенброка)

Получили график, изображенный на рис. 8 справа: подозрительной является та область, где линии уровня не прорисованы, но *гарантировать*, что там есть минимум, нельзя. Попробуем увеличить количество линий уровня (применив опцию `contours`):

```
> plot3d(f(x,y),x=-3..3,y=-3..10,axes=framed,
style=patchcontour,contours=200);
```

Вся поверхность оказалась покрытой линиями уровня, и более светлой осталась плоская область на дне оврага (рис. 9 слева). Команда

```
> contourplot(f(x,y),x=-3..3,y=-3..10,
axes=framed,contours=400);
```

позволяет получить двумерное изображение дна оврага (рис. 9 справа). Чтобы получить представление о том, насколько «круты» стенки оврага, можно применить опцию `view`, которая ограничивает видимую часть графика по оси Oz (рис.10):

```
> plot3d(f(x,y),x=-2..2,y=-2..5,axes=framed,
style=patchcontour,view=0..100);
```

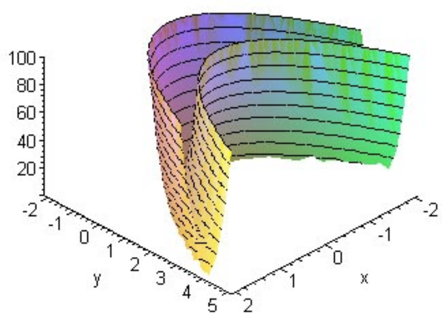


Рис. 10. Стенки оврага (функция Розенброка)

Построим теперь отдельно замкнутую линию уровня, внутри которой находится точка экстремума. Для этого воспользуемся командой `implicitplot`:

```
> implicitplot(f(x,y)=0.1,x=0.2..1.8,y=0.2..2.3,
grid=[150,150],color=black);
```

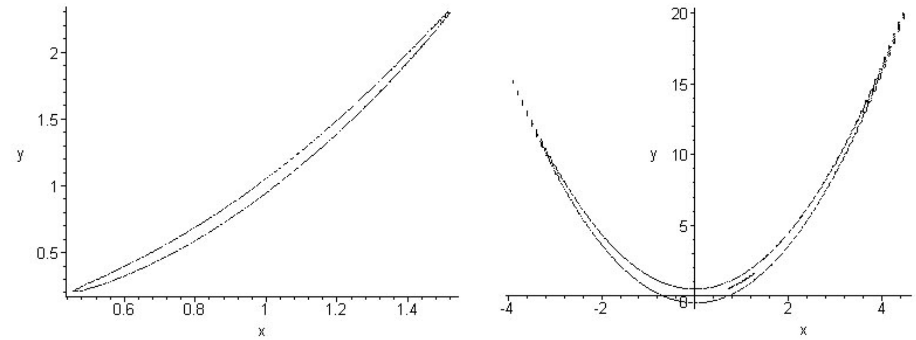


Рис. 11. Линии уровня функции Розенброка

Получили замкнутую кривую $f(x, y) = 0.1$, внутри которой и находится точка минимума (рис. 11 слева). Чтобы лучше представить себе вид дна оврага, можно нарисовать еще несколько линий уровня. Мы ограничимся двумя кривыми – $f(x, y) = 0.1$ и $f(x, y) = 25$:

```
> a1:=implicitplot(f(x,y)=0.1,x=0.2..1.8,y=0.2..2.3,
grid=[150,150],color=black):
> a2:=implicitplot(f(x,y)=25,x=-5..10,y=-1..20,
grid=[150,150],color=black):
> display({a1,a2});
```

В первых двух строчках создаются, но *не выводятся на печать* графические структуры (обратите внимание на двоеточие в конце этих команд). Команда `display` выводит обе кривые на одном графике. Теперь (рис. 11 справа) видно, что овраг изогнут, имеет очень крутые стенки, причем его концы очень узкие (видно, как некорректно работает на этих участках функция `implicitplot`).

Интересно применить программу приближенного нахождения минимума, приведенную в конце предыдущего параграфа, для функции Розенброка с различными начальными приближениями. Как правило, число итераций будет весьма велико, причем основное движение к точке минимума будет происходить по дну оврага с очень низкой скоростью.

4. Условный экстремум

Рассмотрим следующую задачу: найти экстремумы функции $f = f(x)$ на множестве $Q \subset \mathbb{R}^n$, которое задается конечным числом равенств $Q = \{x : g_j(x) = 0, j = 1, \ldots, m\}$. Множество Q называется допустимым, а условия $g_j(x) = 0$ означают, что переменные $x_1, \ldots, x_n$ подчинены некоторым связям.

Точка $x_0 \in Q$ называется точкой локального минимума на множестве Q, если найдется такая окрестность $U(x_0)$, что в любой точке $x \in Q \cap U(x_0)$ функция f принимает значение, не меньшее, чем в точке x_0: $f(x) \geq f(x_0)$. Аналогично определяются точки локального максимума на множестве Q.

Общий метод решения таких задач на *условный* экстремум дается известным *правилом множителей Лагранжа*. Напомним теоремы о необходимых и достаточных условиях.

Теорема (необходимое условие экстремума первого порядка). *Пусть функция f в точке x_0 имеет локальный экстремум на множестве Q. Пусть функции $f, g_1, \ldots, g_m : \mathbb{R}^n \to \mathbb{R}$ непрерывно дифференцируемы в некоторой окрестности точки x_0, причем векторы $g_1'(x_0), \ldots, g_m'(x_0)$ линейно независимы. Тогда существуют числа $\lambda_1, \ldots, \lambda_m$ (множители Лагранжа), такие, что функция Лагранжа $L(\lambda, x) = f(x) + \sum\limits_{i=1}^{m} \lambda_i g_i(x)$ удовлетворяет условиям $\frac{\partial L(\lambda, x_0)}{\partial x_i} = 0$, $i = 1, \ldots, n$.*

Теорема (условия экстремума второго порядка). *Пусть функции $f, g_1, \ldots, g_m : \mathbb{R}^n \to \mathbb{R}$ дважды непрерывно дифференцируемы в некоторой окрестности точки x_0, причем векторы $g_1'(x_0), \ldots, g_m'(x_0)$ линейно независимы.*

1. Если функция f в точке x_0 имеет локальный минимум на множестве Q, то имеет место утверждение предыдущей теоремы, причем второй дифференциал функции Лагранжа

$$d^2 L = \sum_{i,j=1}^{n} \frac{\partial^2 L(\lambda, x_0)}{\partial x_i \partial x_j} dx_i dx_j,$$

где $dx_1, \ldots, dx_n$ подчинены условиям

$$dg_k(x_0) = \sum_{i=1}^{n} \frac{\partial g_k(x_0)}{\partial x_i} dx_i = 0, \; k = 1, \ldots, m,$$

неотрицателен. Если f в x_0 имеет локальный максимум на Q, то второй дифференциал при указанных условиях неположителен.

2. Пусть числа $\lambda_1, \ldots, \lambda_m$ таковы, что имеет место утверждение предыдущей теоремы, и второй дифференциал d^2L при указанных выше условиях строго положителен, если $dx \neq 0$. Тогда функция f в точке x_0 имеет локальный минимум на множестве Q. Если d^2L при указанных выше условиях строго отрицателен, то функция f в точке x_0 имеет локальный максимум на множестве Q.

Часто бывает полезно иметь в виду следующий факт.

Теорема (Вейерштрасса). *Непрерывная функция на компактном множестве достигает своих минимума и максимума.*

Мы ограничимся рассмотрением задачи на условный экстремум для функции двух переменных при наличии одного ограничения. Зададим нашу функцию и ограничения:

```
> f:=(x-2)^2+(y-1)^2;
> g:=x^2+y^2-1;
```

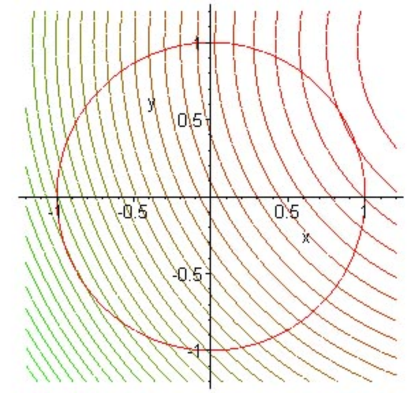


Рис. 12. Линии уровня функции $f(x, y)$ и множество $g(x, y) = 0$

Изобразим на плоскости xOy линии уровня функции $f(x, y)$ и множество $Q = \{(x, y) : \ g(x, y) = 0\}$ (окружность):

```
> with(plots):
> q1:=implicitplot(g(x,y)=0,x=-1..1,y=-1..1,
scaling=constrained):
> q2:=contourplot(f(x,y),x=-1.2..1.2,
```

```
y=-1.2..1.2,scaling=constrained, filled=false,
coloring=[red,green],contours=30):
> display(q1,q2);
```

Опция `scaling=constrained` означает, что масштабы по осям Ox и Oy на графике будут выбраны одинаковыми. Этот фрагмент программы выдает рисунок, приведенный на рис. 12. Полученный график позволяет локализовать точки минимума и максимума, которые являются точками касания окружности $g(x, y) = 0$ и линий уровня функции $f(x, y)$.

Найдем экстремальные точки нашей функции аналитически. Зададим функцию Лагранжа

```
> L:=f+lambda*g;
```

и составим систему

$$\frac{\partial L(\lambda, x, y)}{\partial x} = 0, \quad \frac{\partial L(\lambda, x, y)}{\partial y} = 0, \quad g(x, y) = 0. \qquad (*)$$

```
> Sys_Eqn:={};
> for t in x,y,lambda do
Sys_Eqn:=Sys_Eqn union {diff(L,t)=0}; od;
```

Этот фрагмент программы означает следующее: вначале мы задаем систему из пустого множества уравнений, а затем в цикле добавляем в нее (с помощью команды `union`, которая означает объединение двух множеств) три уравнения: `diff(L,x)=0`, `diff(L,y)=0` и `diff(L,lambda)=0` (нетрудно видеть, что третье уравнение совпадает с уравнением $g(x, y) = 0$). Получаем такую систему:

$$2x - 4 + 2\lambda x = 0, \; 2y - 2 + 2\lambda y = 0, \; x^2 + y^2 - 1 = 0. \qquad (**)$$

Чтобы решить ее, попытаемся воспользоваться командой `solve`:

```
> solve(Sys_Eqn,{x,y,lambda});
```

В этой команде можно, вообще говоря, и не указывать набор переменных, относительно которых решается система. В ответе появляется команда $RootOf$, которая, как мы уже знаем, означает, что корни не могут быть найдены в квадратурах. Применяя команду `evalf(%)`, получаем один ответ:

$$y = 0.4472135956, \quad x = 0.8944271912, \quad \lambda = 1.236067978.$$

Для того, чтобы далее использовать эти значения переменных, удобно применить команду

```
> assign(%);
```

Эта команда *сопоставляет* переменным x, y и λ их значения, полученные при выполнении предыдущей команды, т. е. команды `solve`. Теперь, например, по команде

```
> lambda;
```

мы получим значение 1.236067978 этой переменной.

Является ли полученное решение единственным решением системы? Попробуем воспользоваться командой `fsolve` с указанием пределов изменения переменных. Например, предположим, что есть решение, для которого $\lambda < 0$:

```
> x:='x';y:='y';lambda:='lambda';
> fsolve(Sys_Eqn,{x,y,lambda},{lambda=-infinity..0});
```

Заметим, что в команде `fsolve` приходится указывать набор переменных, относительно которых решается система, если для каких-то из этих переменных указываются пределы изменения. Напомним, что первая строчка этого фрагмента восстанавливает x, y и λ как переменные (после того, как по команде `assign` им были присвоены конкретные значения). Получили второе решение:

$$y = -0.4472135956, \quad x = -0.8944271912, \quad \lambda = -3.236067977.$$

Чтобы сохранить оба набора решений, можно поступить так: определить списки X, Y и Λ:

```
> solve(Sys_Eqn,{x,y,lambda});
> evalf(%);
> assign(%);
> X[1]:=x;Y[1]:=y;Lambda[1]:=lambda;
> x:='x';y:='y';lambda:='lambda';
> fsolve(Sys_Eqn,{x,y,lambda},lambda=-infinity..0);
> assign(%);
> X[2]:=x;Y[2]:=y;Lambda[2]:=lambda;
```

Другой способ найти два решения системы следующий: в каком-нибудь уравнении системы заменить целое число на «число с точкой», например, так:

```
> g:=x^2+y^2-1.;
```

Тогда по команде `solve`(`Sys_Eqn`) получим сразу два решения:

$$x = -0.8944271910,\ y = -0.4472135955,\ \lambda = -3.236067978;$$

$$x = 0.8944271910,\ y = 0.4472135955,\ \lambda = 1.236067978.$$

Чтобы сохранить их для дальнейшего использования, можно поступить так:

```
> s:=solve(Sys_Eqn,{x,y,lambda});
> for i from 1 to 2 do assign(s[i]);X[i]:=x;Y[i]:=y;
Lambda[i]:=lambda;x:='x';y:='y';lambda:='lambda' od;
```

В этом фрагменте программы переменная `s` – это множество, состоящее из двух решений `s[1]` и `s[2]`, а команда `assign` применяется в цикле к каждому из них.

Итак, мы получили два различных решения системы уравнений. Имеет ли эта система другие решения? Если да, то как их найти? Очевидно, что можно выразить λ из первого уравнения системы $(*)$, затем выразить λ из второго уравнения и приравнять эти величины. Полученное уравнение

$$\frac{\partial f(x,y)}{\partial x}\,\frac{\partial g(x,y)}{\partial y} - \frac{\partial f(x,y)}{\partial y}\,\frac{\partial g(x,y)}{\partial x} = 0$$

вместе с уравнением $g(x, y) = 0$ составит систему из двух уравнений с двумя неизвестными. Решение таких систем мы обсуждали в параграфе 2. Однако очевидно, что *в данном конкретном случае* удобно решать систему (**) следующим образом: выразить x через λ из первого уравнения и y через λ из второго уравнения, и подставить эти выражения в третье уравнение системы. Получим уравнение $(1 + \lambda)^2 = 5$, откуда следует, что $\lambda = \pm\sqrt{5} - 1$. Теперь можно, задавая поочередно $\lambda = \sqrt{5} - 1$ и $\lambda = -\sqrt{5} - 1$, составить систему двух уравнений `diff(L,x)=0`, `diff(L,y)=0` (эти уравнения линейные) и решить ее:

```
> Sys_Eqn1:=Sys_Eqn minus {diff(L,lambda)=0};
> for lambda in {sqrt(5)-1,-sqrt(5)-1} do
solve(Sys_Eqn1); od;
```

В первой строке этого фрагмента используется команда `minus`, которая означает «теоретико-множественное вычитание».

Итак, мы показали, что система (**) имеет два решения, которым соответствуют две критические точки нашей функции. Теперь воспользуемся теоремой Вейерштрасса: множество $Q = \{(x, y) : g(x, y) = 0\}$, очевидно, компактно (это окружность), а функция f непрерывна. Значит, она имеет и точки минимума, и точки максимума на множестве Q. Значит, одна из найденных нами точек – минимум, а вторая – максимум. Чтобы проверить, какая из них минимум, а какая – максимум, достаточно просто подставить их в функцию.

В данном случае оказалось, что задачу нахождения экстремумов можно решить, не применяя условия второго порядка. Однако для полноты изложения покажем, как применить эти условия. Вначале построим второй дифференциал функции Лагранжа. Можно действовать так:

```
> DDL:=diff(L,x$2)*dx^2+2*diff(L,x,y)*dx*dy
+diff(L,y$2)*dy^2;
> diff(g,x)*dx+diff(g,y)*dy=0;
> dy:=solve(%,dy);
> DDL;
```

Этот фрагмент программы означает следующее. Вначале мы задаем функцию DDL – второй дифференциал функции L – как функцию четырех переменных: x, y, dx, dy. Далее находим связь между дифференциалами dx и dy. Она выражается равенством $dg(x,y) = \frac{\partial g(x,y)}{\partial x}dx + \frac{\partial g(x,y)}{\partial y}dy = 0$. Из этого равенства выражаем dy через dx (с помощью команды `solve`). Теперь команда `DDL` возвращает функцию DDL от трех переменных: x, y и dx (выражение для dy уже подставлено):

$$(2+2\lambda)dx^2 + \frac{(2+2\lambda)x^2dx^2}{y^2}.$$

Для удобства можно привести подобные (найти коэффициенты при dx^2) с помощью команды `collect`:

```
> collect(%,dx);
```

Тогда второй дифференциал принимает вид

$$\left(2+2\lambda+\frac{(2+2\lambda)x^2}{y^2}\right)dx^2.$$

Найдем его значения в «подозрительных» точках:

```
> for i from 1 to 2 do x:=X[i];y:=Y[i];
lambda:=Lambda[i];print('DDL'=DDL); od;
```

Получаем, что при $x = 0.8944271910$, $y = 0.4472135955$ второй дифференциал положителен (т. е. этому значению соответствует минимум функции f), а при $x = -0.8944271910$, $y = -0.4472135955$ – отрицателен (т. е. этому значению соответствует максимум f).

Полный текст программы MAPLE см. в приложении, стр.44.

5. Задачи с ограничениями типа неравенств

Кратко рассмотрим экстремальную задачу *с ограничениями типа неравенств*: найти точки экстремума функции $f(x)$ на множестве

$\widetilde{Q} = \{x : g_j(x) \leq 0, j = 1, \ldots, m\} \subset \mathbb{R}^n$. Такие задачи относятся к *задачам математического программирования.* Для их решения также применяется метод множителей Лагранжа. В частности, справедлив аналог *необходимого* условия минимума первого порядка с такой оговоркой: в точках локального *минимума* функции f множители Лагранжа $\lambda_1, \ldots, \lambda_m$ должны быть *неотрицательны.* Достаточные условия экстремума могут быть получены для специальных классов функций, например, выпуклых или вогнутых.

Обсудим применение MAPLE для графического нахождения точек экстремума в задачах математического программирования. Пусть, например, по-прежнему требуется найти минимум функции $f(x, y) = (x-2)^2 + (y-1)^2$, но теперь на множестве $\widetilde{Q} = \{x, y : g(x, y) = x^2 + y^2 - 1 \leq 0\}$. Это множество представляет собой круг. Нарисуем линии уровня функции f на множестве $\widetilde{Q}$:

```
>with(plots):
>contourplot(f(x,y),x=-1..1,y=-sqrt(1-x^2)..sqrt(1-x^2),
scaling=constrained,filled=true,coloring=[pink,green]);
```

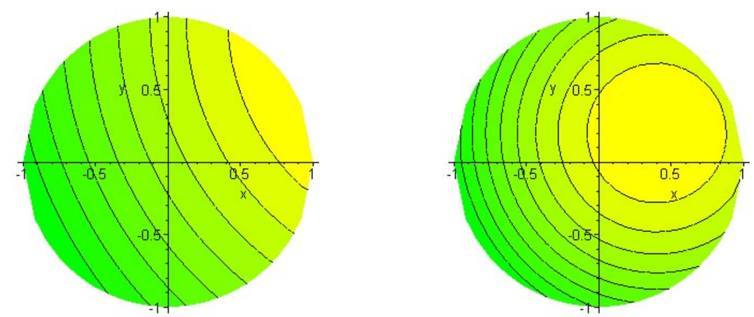


Рис. 13. Линии уровня функции $f(x, y)$ и множество $g(x, y) \leq 0$

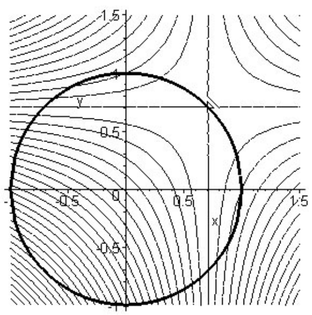
В формате команды `contourplot` допускается границы диапазонов для второй переменной выбирать в виде функций от первой переменной. Мы выбрали промежуток изменения y так, чтобы пара переменных (x, y) пробежала круг $\widetilde{Q}$. По этой команде получаем график, приведенный на рис. 13 слева.

На графике видно, что внутри области $\widetilde{Q}$ функция не имеет экстремумов. Значит, она имеет экстремумы на границе области. Рассмотрим еще функцию

```
> f:=(x,y)->(x-2)^2+(y-1)^2+4*(x^2+y^2);
```

на том же самом множестве $\widetilde{Q}$. На рис. 13 справа изображены линии уровня этой функции. Как следует из графика, эта функция имеет локальный минимум внутри множества $\widetilde{Q}$. Очевидно, локальные экстремумы внутри множества являются и точками безусловного локального экстремума. Задача их нахождения обсуждалась в параграфе 2.

Замечание. Итак, точки экстремума функции $f(x,y)$ на множестве $\widetilde{Q} = \{(x,y) : g(x,y) \le 0\}$ могут быть расположены либо внутри множества (тогда задача по существу эквивалентна задаче нахождения локального безусловного экстремума), либо на его границе $Q = \{(x,y) : g(x,y) = 0\}$. Но это не означает, что в последнем случае задача сводится к задаче нахождения экстремума с ограничениями типа равенств! Действительно, точка экстремума для функции $f(x,y)$ на множестве Q может *не быть экстремальной точкой* этой же функции на множестве $\widetilde{Q}$.



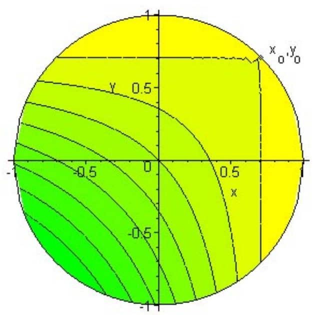


Рис. 14. Линии уровня функции $f(x,y)$ и множество $g(x,y) \le 0$

В качестве примера рассмотрим функцию $f=(x-\frac{\sqrt{2}}{2})(y-\frac{\sqrt{2}}{2})$ на круге $\widetilde{Q} = \{(x,y) : x^2+y^2-1 \le 0\}$ и на окружности $Q = \{(x,y) : x^2+y^2-1 = 0\}$. Линии уровня показаны на рис. 14. Очевидно, в точке $(x_0,y_0) = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}) \in Q$ эта функция имеет критическую точку – седло. Из этой точки в направлении вдоль окружности функция f убывает, следовательно, на множестве Q она имеет локальный максимум в точке (x_0, y_0). Однако вдоль прямой $y = x$ в направлении внутрь круга функция f растет. Следовательно, (x_0, y_0) не является экстремальной точкой функции f на множестве $\widetilde{Q}$.

***

Для численного решения задач математического программирования разработан так называемый *метод штрафных функций*. Дадим краткое описание основных идей этого метода в простейшем варианте. Рассмотрим задачу минимизации функции $f(x) : \mathbb{R}^n \to \mathbb{R}$ на некотором множестве $\widetilde{Q} \subset \mathbb{R}^n$. Наряду с исходной задачей рассмотрим последовательность задач *безусловной* минимизации некоторых вспомогательных функций $\Phi_m(x)$. При этом подберем функции $\Phi_m(x)$ так, чтобы при больших m они мало отличались от функции $f(x)$ на множестве $\widetilde{Q}$ и быстро возрастали на множестве $\mathbb{R}^n \backslash \widetilde{Q}$. Тогда при больших m из-за того, что функция $\Phi_m(x)$ вне множества $\widetilde{Q}$ принимает достаточно большие значения, можно надеяться, что минимальное значение $\Phi_m(x)$ будет достигаться на множестве $\widetilde{Q}$, причем в точке, близкой к решению исходной задачи. Другими словами, задачу условной минимизации предлагается аппроксимировать задачей безусловной минимизации для новой функции $\Phi_m(x)$ при достаточно большом значении m.

Функцию $\Phi_m(x)$ можно искать в виде $\Phi_m(x) = f(x) + P_m(x)$, где $P_m(x)$ – так называемая «штрафная функция».

Определение. Рассмотрим задачу минимизации функции $f(x) : \mathbb{R}^n \to \mathbb{R}$ на некотором множестве $\widetilde{Q} \subset \mathbb{R}^n$. Последовательность *неотрицательных* функций $P_m(x) : \mathbb{R}^n \to \mathbb{R}$ называется *штрафной функцией* или *штрафом*[5], если

$$\lim_{m\to\infty} P_m(x) = \begin{cases} 0, & x \in \widetilde{Q}, \\ +\infty, & x \in \mathbb{R}^n \backslash \widetilde{Q}. \end{cases}$$

Название *штраф* означает, что при больших m за нарушение условия $x \in \widetilde{Q}$ приходится «платить большой штраф» в размере $P_m(x)$.

[5] Иногда для удобства штрафом называют не только последовательность $\{P_m(x)\}_{m=1}^{\infty}$, но и каждую функцию $P_m(x)$ в отдельности.

Очевидно, для каждого множества существует бесконечно много способов построить штраф. Мы ограничимся таким примером: пусть $\widetilde{Q} = \{x : g(x) \le 0\}$, тогда штраф можно выбрать так:

$$P_m(x) = \begin{cases} 0, & g(x) \le 0, \\ m \cdot g(x)^2, & g(x) > 0. \end{cases}, \quad m = 1, 2, \ldots$$

Такой способ выбора штрафа хорош тем, что если функция $g(x)$ гладкая, то и функция $P_m(x)$ тоже будет гладкая.

Заметим, что для задачи минимизации функции f на множестве $Q = \{x : g(x) = 0\}$ тоже можно применить метод штрафов, но только штраф выбрать, например, так: $P_m(x) = m \cdot g(x)^2$.

Приведем программу, реализующую метод штрафов в случае минимизации функции $f(x, y) = (x-2)^2 + (y-1)^2$ на множестве $\widetilde{Q} = \{(x, y) : g(x, y) = x^2 + y^2 - 1 \le 0\}$, которая представляет собой вариант программы на стр. 23. Здесь штрафная функция записана как $P_m(x, y) = m \cdot \max\{g(x, y), 0\}^2$.

```
> f:=(x,y)->(x-2)^2+(y-1)^2;
> g:=(x,y)->x^2+y^2-1;
> Phi:=(x,y,m)->f(x,y)+m*(max(g(x,y),0))^2;
> Phi_x:=piecewise(g(x,y)<=0,diff(f(x,y),x),
g(x,y)>0,diff(f(x,y)+m*g(x,y)^2,x));
> Phi_y:=piecewise(g(x,y)<=0,diff(f(x,y),y),
g(x,y)>0,diff(f(x,y)+m*g(x,y)^2,y));
> x0:=1.:y0:=0.:List[0]:=[x0,y0]:
> w:=1:j:=0:m:=500:
> while w>0.000001 do x:=x0:y:=y0:a:=1:
for i from 1 to 20 do x1:=x0-a*Phi_x:y1:=y0-a*Phi_y:
if Phi(x1,y1,m)-Phi(x0,y0,m)<0 then break
else a:=a/2:fi:od:
w:=sqrt((x1-x0)^2+(y1-y0)^2):
x0:=x1:y0:=y1:j:=j+1:List[j]:=[x0,y0]:od:
> j;x0;y0;
> with(plots):with(plottools):
> for k from 0 to j do
d[k]:=display(POINTS(List[k]),color=red): od:
```

```
> q:=display(seq([d[k]],k=0..j),insequence=true):
> for k from 1 to j do
c[k]:=display(line(List[k-1],List[k])): od:
> display(q,seq([c[k]],k=1..j));
```

Итак, в этой программе применяется градиентный метод для поиска минимума функции $\Phi(x,y) = f(x,y)+m\cdot\max\{g(x,y),0\}^2$ при $m = 500$. Поскольку MAPLE плохо справляется с дифференцированием функций, заданных с помощью операций `max` и `min`, градиент функции $\Phi(x,y)$ определяется здесь как кусочно заданная функция (командой `piecewise`): внутри круга $g(x,y) \leq 0$ он равен градиенту функции f, а вне круга – градиенту функции $f + m \cdot g^2$. В данном случае за 2296 итераций получим ответ – точку (0.8947261635, 0.4479944917), а также визуализацию и анимацию приближений.

6. Стандартные средства MAPLE для нахождения экстремумов

В системе MAPLE определены стандартные функции, позволяющие находить глобальные экстремумы и экстремумы в областях: `minimize`, `maximize`, `extrema`. При этом форма записи этих команд и результаты, которые могут быть получены, существенно зависят от версии MAPLE. Приведем примеры их применения:

```
> minimize(f); maximize(f,location);
> minimize(f,x=-2..2); maximize(f,x=-2..2,location);
> extrema(f,{ }); extrema(f,{ },x,'s');s;
```

Если указана опция `location`, то будет выведена и экстремальная точка, а иначе – только экстремальное значение функции. Заметим, что эти функции, вообще говоря, не находят *локальные* экстремумы. Например, для функции $f = \frac{1}{4}x^4 - \frac{2}{3}x^3 - \frac{3}{2}x^2 + 2$, которая рассматривалась в §1, команда `minimize(f,x=-2..2,location)` выдаст минимум в точке $x = 2$, а команда `maximize(f,x=-2..2,location)` выдаст максимум в точке $x = -2$ (т. к. максимум и минимум *на отрез-*

ке $[-2,2]$ действительно достигаются на его границе). Команда `minimize(f,location)` выдаст глобальный минимум (в точке $x=3$), а по команде `maximize(f,location)` получим, что глобальный максимум функции равен ∞, и достигается он в точках $+\infty$ и $-\infty$.

Команда `extrema(f,{ })` выдаст глобальные (на всей оси) экстремальные значения 2 и $-\frac{37}{4}$. Наконец, по команде `extrema(f,{ },x,'s')` в переменной `s` будет сформирован список экстремальных точек (в данном случае это будет множество из трех элементов $\{\{x=3\},\{x=0\},\{x=-1\}\}$).

Эти команды можно применять и для функций нескольких переменных. Например, по команде

```
> minimize(f(x,y),location);
```

для функции Розенброка $f=100(y-x^2)^2+(1-x)^2$, которая рассматривалась в §3, получим

$$0,\{[\{x=1,y=1\},0]\}.$$

Это означает, что в точке $x=1$, $y=1$ функция достигает минимума, причем минимальное значение функции равно 0.

Более того, команду `extrema` можно применять для поиска условного экстремума функции многих переменных. Например, для функции $f=(x-2)^2+(y-1)^2$ и ограничения $g:=x^2+y^2-1$, которые рассматривались в §4, применяем команду

```
> extrema(f,g=0);
```

MAPLE выдает ответ, содержащий функцию $RootOf$. Заменим 0 в уравнении $g(x,y)=0$ на «ноль с точкой»

```
> extrema(f,g=0.);
```

и получим два экстремальных значения

$$1.527864045,\ 10.47213596.$$

Для нахождения экстремальных точек команда `extrema` применяется в следующем варианте:

```
> extrema(f,g=0.,{x,y},'s');s;
```

7. Задания для самостоятельной работы

1. Найдите локальные экстремумы функции одной переменной. Если функция имеет бесконечное количество экстремальных точек, локализуйте и найдите 3-4 из них.

№	Функция $f(x)$
1	$f = \sin(10x) \cdot e^{x^2} + x^3$
2	$f = \sin(5x) \cdot e^{0.1x} + x^3 + \cos(4x)$
3	$f = x^6 + 101x^5 + 425x^4 - 425x^2 - 101x - 1$
4	$f = x^5 - 5x^4 - 4x^3 + 20x^2 - 5x + 25$
5	$f = x^6 - 24x^5 - 375x^4 + 375x^2 + 24x - 1$
6	$f = \sin(5x) \cdot e^{x^2/2} + x^2$
7	$f = \cos(5x) \cdot e^{x} + x^3 - 10\sin(2x)$
8	$f = \frac{1}{4}x^4 + \sin(2x)^2 + 0.1x^2 - x^2\sin(2x) - 0.3\sin(2x) + \frac{1}{99}$
9	$f = x^4 - 6x^2\cos(x) + 5(\cos(x))^2$
10	$f = x^4 - 6x^2\cos(x) + 5\cos(x^2)$

2. Найдите локальные экстремумы функции двух переменных. Если функция имеет бесконечное количество экстремальных точек, локализуйте и найдите 2-3 из них.

№	Функция $f(x, y)$
1	$f = -2e^{-(x-2)^2-y^2} + e^{-(x+2)^2-y^2}$
2	$f = -3e^{-x^2-y^2} - 2e^{-(x-2)^2-y^2}$
3	$f = -3e^{-x^2-y^2} - 2e^{-(x-2)^2-y^2} + e^{-x^2-(y-2)^2}$
4	$f = 10(y - \sin x)^2 + 0.1x^2$
5	$f = (1 + \sin^2 x)(1 + \sin^2 y)$

№	Функция $f(x,y)$
6	$f=e^{x^2+y^2}+2x-3.5y$
7	$f=-x^2\cdot e^{1-x^2-20.25(x-y)^2}$
8	$f=x^4+(y+1)^4-2x^2+2(y+1)^2+4x(y+1)$
9	$f=10x-y+e^{x^2+2y^2}$
10	$f=4x-1.5y+e^{0.16x^2+1.4y^2}$

3. Найдите локальные экстремумы функции двух переменных $f(x,y)$ на множестве $Q=\{(x,y):g(x,y)=0\}$ и на множестве $\widetilde{Q}=\{(x,y):g(x,y)\le 0\}$.

№	Функция $f(x,y)$	Функция $g(x,y)$
1	$f=(x-2)^2+y^2$	$g=x^2-y^3$
2	$f=(x-2)^2+y^2$	$g=x^3-y^2$
3	$f=(x-2)^2+y^2$	$g=x^3-y^2+x$
4	$f=(x-2)^2+y^2$	$g=y^2+\frac{1}{2}x^2-1$
5	$f=y^2+\frac{1}{2}x^2$	$g=(x-2)^2+y^2-1$
6	$f=(x-2)^2+y^2+\sin(xy)$	$g=y^2+\frac{1}{2}x^2-5$
7	$f=(x+y+2)^2+5y^2$	$g=y^2+0.1(x-0.3y)^2-5$
8	$f=(x+y+2)^2-5y^2$	$g=y^2+0.1(x-0.3y)^2-5$
9	$f=(x+y+2)^2+5y^2+$ $+0.5\Big(y^2+10(x-y)^2\Big)$	$g=y^2+0.1(x-0.3y)^2-5$
10	$f=(x+y)^3+5y^4$	$g=y^2+0.1(x-4y)^2-5$

Приложение А. Тексты программ MAPLE

К пункту 1. Нахождение экстремума функции одной переменной.

```
> restart;
> f:=1/4*x^4-2/3*x^3-3/2*x^2+2;
> plot(f,x=-2..4);
> solve(diff(f,x)=0);
> x:=0:print('x'=x, 'f'=f);
> x:=3:print('x'=x, 'f'=f);
> x:=-1:print('x'=x, 'f'=f);
> x:='x';
> ddf:=diff(f,x$2);
> x:=0:print('x'=x,'sign(ddf)'=sign(ddf));
> x:=3:print('x'=x,'sign(ddf)'=sign(ddf));
> x:=-1:print('x'=x,'sign(ddf)'=sign(ddf));
```

К пункту 2. Нахождение экстремума функции двух переменных.

```
> restart;
> f:=(x,y)->x^4+y^4-2*x^2-2*y^2+4*x*y;
> plot3d(f(x,y),x=-2.5..2.5,y=-2.5..2.5,axes=frame);
> with(plots):
> contourplot3d(f(x,y),x=-2.5..2.5,y=-2.5..2.5,
axes=frame,filled=true,contours=25);
> contourplot(f(x,y),x=-2.5..2.5,y=-2.5..2.5,
axes=frame,filled=true,contours=25);
> Sys_Eqn:=diff(f(x,y),x)=0,diff(f(x,y),y)=0;
> solve({Sys_Eqn});
> evalf(%);
> implicitplot({Sys_Eqn},x=-2.5..2.5,y=-2.5..2.5);
> fsolve({Sys_Eqn},{x=0.1..2.5,y=-2.5..0.1});
> fsolve({Sys_Eqn},{x=-0.05..0.05,y=-0.05..0.05});
> fsolve({Sys_Eqn},{x=-2.5..-0.1,y=0.1..2.5});
> subs(x=sqrt(2),y=-sqrt(2),Sys_Eqn);
> subs(x=-sqrt(2),y=sqrt(2),Sys_Eqn);
> My_matr:=matrix(2,2,
[[diff(f(x,y),[x,x]),diff(f(x,y),[x,y])],
[diff(f(x,y),[y,x]),diff(f(x,y),[y,y])]]);
```

```
> with(linalg):d:=det(My_matr);
> My_List:=[[0,0],[sqrt(2),-sqrt(2)],
[-sqrt(2),sqrt(2)]];
> for z in My_List do x:=z[1];y:=z[2];print('x'=x,'y'=y,
'First_minor'=My_matr[1,1],'Second_minor'=d) od:
> f(sqrt(2),-sqrt(2)); f(-sqrt(2),sqrt(2));
> f1:=(x)->f(x,x);
> f2:=(x)->f(x,-x);
> plot(f1,-0.5..0.5);
> plot(f2,-0.5..0.5);
```

К пункту 4. Нахождение условного экстремума функции двух переменных.

```
> restart;
> f:=(x-2)^2+(y-1)^2;
> g:=x^2+y^2-1;
> L:=f+lambda*g;
> Sys_Eqn := {};
> for t in x,y,lambda do
Sys_Eqn:=Sys_Eqn union {diff(L,t)=0}; od;
> solve(Sys_Eqn);
> evalf(%);
> assign(%);
> X[1]:=x;Y[1]:=y;Lambda[1]:=lambda;
> x:='x';y:='y';lambda:='lambda';
> fsolve(Sys_Eqn,{x,y,lambda},lambda=-infinity..0);
> assign(%);
> X[2]:=x;Y[2]:=y;Lambda[2]:=lambda;
> x:='x';y:='y';lambda:='lambda';
> DDL:=diff(L,x$2)*dx^2+2*diff(L,x,y)*dx*dy
+diff(L,y$2)*dy^2;
> diff(g,x)*dx+diff(g,y)*dy=0;
> dy:=solve(%,dy);
> collect(DDL,dx);
> for i from 1 to 2 do lambda:=Lambda[i];x:=X[i];
y:=Y[i];print('DDL'=DDL); od;
```

Приложение В. Глоссарий

В этом приложении мы перечислим и коротко прокомментируем некоторые команды MAPLE, которые встречались в данном пособии.

`restart` – команда, по которой обнуляются все переменные. Рекомендуется применять в начале программы, стр. 5.

`plot` – вывод двумерных графиков, стр. 6.

`diff` – нахождение производной, стр. 7.

`solve` – решение уравнений и систем уравнений, стр. 7.

`print` – управление выводом, стр. 7.

`x:='x'` – восстановление переменной (в данном случае `x`) после присвоения ей конкретного значения, стр. 8.

`for..from..to` – оператор цикла, стр. 9.

`do..od` – операторные скобки. Команда `od` эквивалентна команде `end do`, стр. 9.

`fsolve` – приближенное решение уравнений и систем уравнений, стр. 9.

`if..then..esle..fi` – условный оператор. Команда `fi` эквивалентна команде `end if`, стр. 9.

`while..do..od` – оператор цикла, стр. 10.

`plot3d` – вывод трехмерных графиков, стр. 11.

`axes` – опция графических команд (например, команды plot3d), определяющая вид осей координат, стр. 11.

`with` – подключение пакетов расширения MAPLE, стр. 12.

`plots` – пакет графического расширения MAPLE, стр. 12.

`contourplot3d` – вывод контурных трехмерных графиков. Требует подключения пакета `plots`, стр. 12.

`filled` – опция графических команд (например, команды `contourplot3d`), задающая «заливку» графика, стр. 12.

`contours` – опция некоторых графических команд (например, команд `plot3d` и `contourplot3d`), определяющая количество контурных линий, стр. 12.

`contourplot` – вывод контурных двумерных графиков. Требует подключения пакета `plots`, стр. 13.

`style` – опция графических команд (например, команды `plot3d`), определяющая стиль линий сетки на графике. Значение `patchcontour` определяет контурный график, а значение `point` – точечный график, стр. 13.

`coloring` – опция графических команд (например, команды `plot`), определяющая цвета раскраски графика, стр. 13.

`evalf` – вычисление приближенного значения, стр. 15.

`%` – результат последнего (по времени) вычисления, стр. 15.

`implicitplot` – вывод графика функции, заданной неявно. Требует подключения пакета `plots`, стр. 15.

`subs` – подстановка выражений, стр. 17.

`matrix` – матричный тип данных. Использует тип данных «список», который задается квадратными скобками, стр. 17.

`linalg` – пакет расширения линейной алгебры, стр. 18.

`det` – нахождение определителя матрицы. Требует подключения пакета `linalg`, стр. 18.

`for..in` – оператор цикла, стр. 19.

`grid` – опция графических команд (например, команды `plot3d`), определяющая количество линий сетки, стр. 19.

`display` – вывод графических структур. Требует подключения пакета `plots`, стр. 23.

`insequence` – опция команды `display`. Значение `true` создает анимационный эффект, стр. 23.

`assign` – сопоставление переменным их значений, стр. 31.

`collect` – приведение подобных, стр. 34.

`scaling` – опция графических команд (например, команды `plot`), определяющая соотношение масштабов по осям. Значение `constrained` означает, что масштабы по осям выбираются равными, стр. 35.

`piecewise` – определение кусочно заданных функций, стр. 38.

`minimize,maximize,extrema` – стандартные средства MAPLE нахождения экстремумов, стр. 39.

Список литературы

1. Васильев Ф. П. Численные методы решения экстремальных задач. – М.:Наука, 1988. – 552 с.

2. Сухарев А. Г., Тимохов А. В., Федоров В. В. Курс методов оптимизации. – М.:Наука, 1986. – 328 с.

3. Плис А. И., Сливина Н. А. Лабораторный практикум по высшей математике: Учебное пособие для втузов. – М.:Высшая школа, 1983. – 208 с.

4. Турчак Л. И, Плотников П. В. Основы численных методов. – М.: Физматлит, 2005. – 300 с.

5. Матросов А. Maple 6. Решение задач высшей математики и механики. – СПб.:BHV – Санкт-Петербург, 2001. – 526 с.

6. Васильев А. Н. Maple 8. Самоучитель. – К.:Диалектика, 2003. – 351 с.

7. Дьяконов В. П. Математическая система MAPLE V R3/R4/R5. – М.:«Солон», 1998. – 399 с.

8. Дьяконов В. П. Maple 6: Учебный курс. – СПб.:Питер, 2001. – 608 с.

9. Дьяконов В. П. Maple 7: Учебный курс. – СПб.:Питер, 2002. – 672 с.

10. Дьяконов В. П. Maple 8 в математике, физике и образовании – М.: СОЛОН-Пресс, 2003. – 656 с.

11. Дьяконов В. П. Maple 9 в математике, физике и образовании – М.: СОЛОН-Пресс, 2004. – 685 с.

12. Дьяконов В. П. Maple 9.5/10 в математике, физике и образовании – М.: СОЛОН-Пресс, 2006. – 720 с.

13. Василевська Ю. В. Розв'язання задач в Maple. Навчальний посібник. – Х.:ХНУ імені В. Н. Каразіна, 2006. – 104 с.

Навчальне видання

Ігнатович Світлана Юріївна
Райхцаум Раїса Борисівна

Застосування системи MAPLE
для розв'язання задач оптимізації

Макет обкладинки І. М. Дончик
Коректор А. І. Меліхова

Підписано до друку 26.09.08. Формат $60 \times 84/16$.
Папір офсетний. Друк ризографічний.
Обл.-вид. арк. 2.46. Умов.-друк. арк. 1.97.
Наклад 100 прим. Ціна договірна

61077, Харків, майдан Свободи, 4,
Харківський національний університет імені В. Н. Каразіна
Видавництво Харківського національного
університету імені В. Н. Каразіна

Надруковано ФОП «Петрова І. В.»
61144, Харків-144, вул. Гв. Широнінців, $79^{\text{В}}$, к.137,
Тел. 362-01-52

Свідоцтво про державну реєстрацію ВОО № 948011 від 03.01.03