

## **Практикум 11. Исследование функций и построение графиков.**

**Цель работы** – освоить исследование функций и построение графиков с использованием производных; изучить возможности среды MatLab для исследования функций; научиться использовать встроенную функцию fzero для нахождения нулей, стационарных точек и точек перегиба функции; познакомиться с опциями оптимизации среды MatLab.

**Продолжительность работы** - 4 часа.

**Оборудование, приборы, инструментарий** – работа выполняется в компьютерном классе с использованием пакета MatLab.

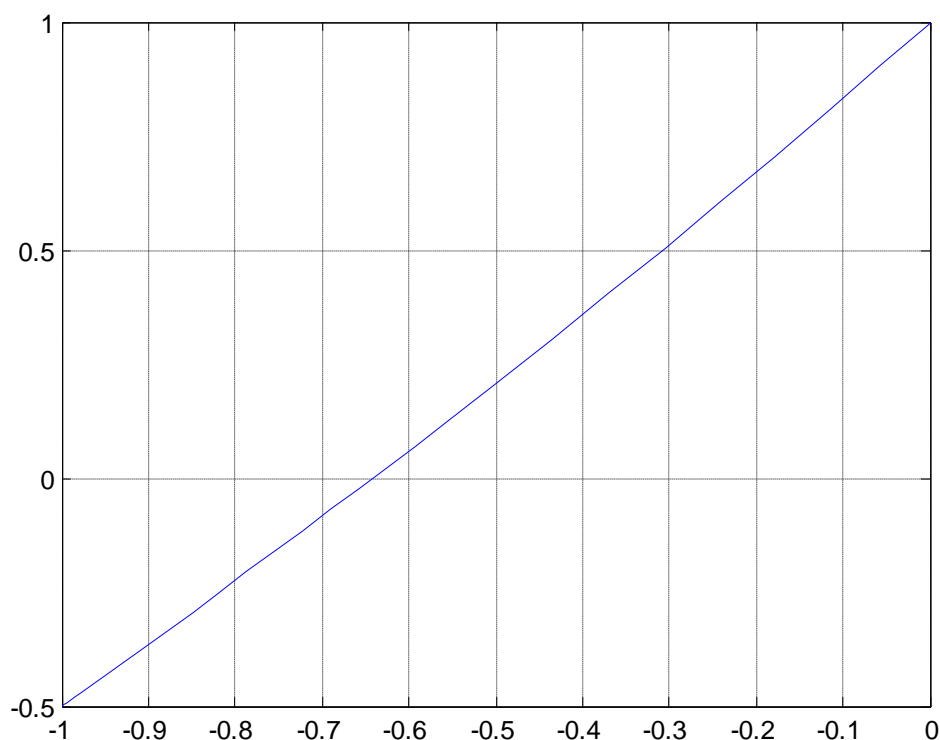
### ***Порядок выполнения***

1. Упражнения выполняются параллельно с изучением теоретического материала.
2. После выполнения каждого упражнения результаты заносятся в отчёт.
3. При выполнении упражнений в случае появления сообщения об ошибке рекомендуется сначала самостоятельно выяснить, чем оно вызвано, и исправить команду; если многократные попытки устранить ошибку не привели к успеху, то проконсультироваться с преподавателем.
4. Дома доделать упражнения из раздела «Краткие теоретические сведения и практические упражнения», которые Вы не успели выполнить во время аудиторного занятия.
5. После выполнения упражнений выполнить дополнительные упражнения для самостоятельной работы и ответить на контрольные вопросы и (см. ниже).
6. Подготовить отчёт, в который включить упражнения из раздела «Краткие теоретические сведения и практические упражнения» и упражнения для самостоятельной работы. Отчёт представить в виде документа Microsoft Word, имя файла (пример): mp\_10\_Ivanov\_P\_01\_s\_1 (факультет\_группа\_Фамилия студента\_Инициал\_номер лабораторной, семестр). Отчет должен содержать по каждому выполненному упражнению: № упражнения, текст упражнения; команды, скопированные из командного окна, с комментариями к ним и результаты их выполнения, включая построенные графики; тексты М-сценариев и М-функций; выводы.

## *Краткие теоретические сведения и практические упражнения*

**1. Нахождение нулей функции.** Встроенная функция *fzero* позволяет приближённо вычислить корень уравнения по заданному начальному приближению. В самом простом варианте *fzero* вызывается с двумя входными и одним выходным аргументом  $x=fzero('myfun', x0)$ , где *myfun* имя файл-функции, вычисляющей левую часть уравнения,  $x0$  – начальное приближение к корню,  $x$  – найденное значение корня. Перед нахождением корней полезно построить график функции, входящей в левую часть уравнения. Имеет смысл воспользоваться *fplot*, построив более точный график по сравнению с *plot*.

**Пример 1.** Найдём приближённое решение уравнения  $2^x + x = 0$ . Так как функция  $2^x + x$  возрастающая и  $f(-1) < 0$ ,  $f(0) > 0$ , то уравнение имеет ровно один корень и этот корень принадлежит промежутку  $[-1; 0]$ . Создадим файл-функцию *myfun* и построим её график на отрезке  $[-1; 0]$ , нанеся сетку.



Видим, что корень находится вблизи точки  $x_0 = -0,5$ .

Уточняем значение корня:

```
x=fzero('myfun',-0.5)
```

Получаем

```
x =
```

```
-0.6412
```

Проверяем подстановкой

```
y=myfun(x);
```

```
y =
```

```
0
```

**Упражнение 1.** Найти корни уравнения  $\sin x = x^2 \cos x$ , принадлежащие промежутку  $[-5;5]$ . Результаты сохранить в текстовом файле.

Важной особенностью **fzero** является то, что она вычисляет только те корни, где функция меняет знак, а не касается оси абсцисс.

**Пример 2.** Попробуем найти корень уравнения  $x^2 = 0$  с помощью **fzero**.

```
>> x1=fzero('x^2',-0.1)
```

```
Exiting fzero: aborting search for an interval containing a sign change
```

```
because NaN or Inf function value encountered during search.
```

```
(Function value at 1.37296e+154 is Inf.)
```

```
Check function or try again with a different starting value.
```

```
x1 =
```

```
NaN
```

В данном случае **fzero** пыталась найти промежуток, на границах которого функция  $y = x^2$  принимала бы значения разных знаков, что, естественно, не удалось, и было выдано сообщение об ошибке.

Вторым аргументом **fzero** может быть также интервал, на концах которого функция принимает значения разных знаков.

**Упражнение 2.** Набрать в командном окне команды `x1=fzero('cos',[-10,10])` и `x1=fzero('sin',[-10,10])`. Объяснить результат.

**Упражнение 3.** Найти все корни уравнения  $\sin x - x^2 \cos x = 0$  на отрезке  $[-10;10]$ . Ответ записать в текстовый файл.

## **2. Опции оптимизации.**

Функции из Optimization Toolbox обычно принимают дополнительный аргумент, который называется options в документации – структуру, содержащую различные параметры оптимизации. Вызов `optimset('Opt1', Val1, 'Opt2', Val2, ...)` создаёт структуру, где поле Opt1 имеет значение Val1, а поле Opt2 – значение Val2, а все остальные поля – значения по умолчанию. Основные поля (остальные поля и подробности можно посмотреть в Optimization Toolbox > User's Guide > Arguments and Options Reference > Optimization Options):

**Display** – Подробность показываемых данных. Это должна быть одна из строк: 'off' (не выводить ничего), 'iter' (выводить значения после каждой итерации), 'notify' (выводить значения при обнаружении расходимости), 'final' (выводить окончательный результат).

**MaxFunEvals** – Максимальное число вычислений функции (положительное целое).

**MaxIter** – Максимальное число итераций (положительное целое)

**TolFun** – Допустимая погрешность в значении функции (положительное действительное число)

**TolX** – Допустимая погрешность в значении аргумента (положительное действительное число)

**FunValCheck** – выдаётся ли ошибка, если оптимизируемая функция принимает комплексные значения или NaN (и для некоторых функций – бесконечность)

**OutputFcn** – функция (или функции), которая вызывается после каждой итерации.

**PlotFcns** -- функция (или функции), график которых строится после каждой итерации.

Пример 3.

```
>> optimset('fzero') %% параметры оптимизации для fzero по умолчанию
```

```
ans =
```

```
Display: 'notify'
```

```
MaxFunEvals: []
```

```
MaxIter: []
```

```
TolFun: []
```

```
TolX: 2.2204e-016
```

```
FunValCheck: 'off'
```

%% покажем промежуточные результаты, ищем корень с точностью 0,0001,  
и строим график значения функции в зависимости от номера шага

```
>> opts = optimset('Display', 'iter', 'TolX', 0.0001, 'PlotFcns', {@optimplotfval});
```

```
>> x=fzero(@tan, 1, opts).
```

%% сначала Matlab ищет интервал рядом с 1, в котором функция меняет знак

%% Func-count – число вызовов функции, а a и b – начало и конец текущего интервала

Search for an interval around 1 containing a sign change:

Func-count	a	f(a)	b	f(b)	Procedure
1	1	1.55741	1	1.55741	initial interval
3	0.971716	1.46458	1.02828	1.65879	search
5	0.96	1.42836	1.04	1.70361	search
7	0.943431	1.37915	1.05657	1.77015	search
9	0.92	1.31326	1.08	1.87122	search
11	0.886863	1.22671	1.11314	2.03031	search
13	0.84	1.11563	1.16	2.2958	search
15	0.773726	0.976924	1.22627	2.78681	search

17	0.68	0.808661	1.32	3.90335	search
19	0.547452	0.609604	1.45255	8.41735	search
21	0.36	0.376403	1.64	-14.427	search

%% а теперь 0 внутри этого интервала

Search for a zero in the interval [0.36, 1.64]:

Func-count	x	f(x)	Procedure
21	0.36	0.376403	initial
22	0.392546	0.414034	interpolation
23	1.01627	1.61461	bisection
24	1.32814	4.03979	bisection
25	1.48407	11.5014	bisection
26	1.64	-14.427	bisection
27	1.63125	-16.5214	interpolation
28	1.59664	-38.6824	bisection
29	1.56203	114.124	bisection
30	1.57934	-117.069	interpolation
31	1.57902	-121.564	interpolation
32	1.5748	-249.827	bisection
33	1.57269	-528.771	bisection
34	1.57163	-1197.05	bisection
35	1.5711	-3252.16	bisection

%% предупреждение, что **скорее всего** найденная точка – не ноль функции,  
а точка разрыва

Current point x may be near a singular point. The interval [0.36, 1.64]

reduced to the requested tolerance and the function changes sign in the interval,

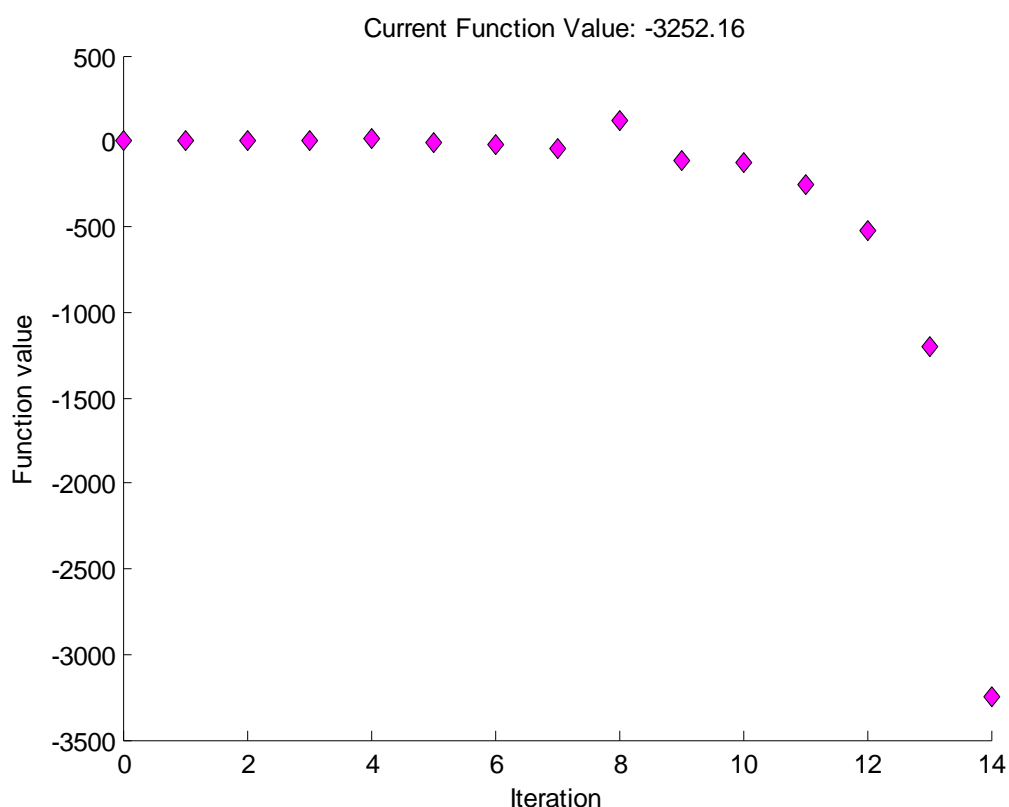
but f(x) increased in magnitude as the interval reduced.

x =

1.5711

y =

-3.2522e+003



### 3. Минимизация функций.

Поиск локального минимума функции одной переменной на некотором отрезке осуществляется с помощью **fminbnd** с тремя входными аргументами: **fminbnd ('fname',x1,x2)** , где x1, x2 – границы отрезка, на котором ищется минимум. При нахождении минимума функции следует сначала построить график с помощью **fplot** и определить, какому промежутку принадлежит точка минимума. Для одновременного вывода точки минимума и значения в ней используется **fminbnd** с двумя выходными аргументами: **[x0,f]= fminbnd ('fname',x1,x2)**. Для нахождения локального максимума нет специальной функции, но очевидно, что локальные максимумы можно найти как минимумы противоположной функции.

**Упражнение 4.** Найти локальные максимум и минимумы для функции  $f(x) = e^{-x} \sin(3\pi x)$  на промежутке  $[0; 2]$ . Ответ записать в текстовый файл.

**3. Нахождение точек перегиба.** Точки перегиба функции можно найти как нули второй производной или как точки экстремумов первой, построив предварительно их графики.

**Упражнение 5.** Найти точки перегиба для функции  $f(x) = e^{-x} \sin(3\pi x)$  на промежутке  $[0; 2]$ .

#### **4. Полное исследование функции.**

**Упражнение 6.** Построить график функции  $f(x) = \frac{x^3 - 3x^2 - 3x}{x^2 - 1}$ . Найти нули функции, точки экстремума и значения в них, точки перегиба, значения в них, значения тангенса угла наклона касательной в точке перегиба, найти односторонние пределы в точках разрыва, уравнения асимптот. Обозначить на графике экстремумы, построить касательные в окрестностях точек перегиба, асимптоты.

#### ***Задания для самостоятельной работы***

1. Выполнить упражнения из раздела «Краткие теоретические сведения и практические упражнения», которые не успели сделать в аудитории.

2. Самостоятельно выполнить упражнения:

**Упражнение С1.** Найти точки перегиба для функции  $f(x) = \sin x - x^2 \cos x$  на промежутке  $[-10; 10]$ .

**Упражнение С2.** Построить график функции  $f(x) = e^{1/(x^2-1)}$ . Найти нули функции, точки экстремума и значения в них, точки перегиба, значения в них, значения тангенса угла наклона касательной в точке перегиба, найти односторонние пределы в точках разрыва, уравнения асимптот. Обозначить на графике экстремумы, построить касательные в окрестностях точек перегиба, асимптоты.

3. Ответить на контрольные вопросы:

1. Можно ли, используя функцию `fzero`, найти нули функции  $f(x) = \cos x - 1$ ?
2. Почему, прежде чем искать нули функции с помощью функции `fzero`, рекомендуется построить график функции?



3. Почему, прежде чем искать минимумы функции с помощью функции `fminbnd`, рекомендуется построить график функции?

*Список рекомендуемой литературы*

1. В.Г.Потемкин "Введение в Matlab" (v 5.3), <http://matlab.exponenta.ru/ml/book1/index.php>.
2. Сборник задач по математике для втузов под ред. А.В.Ефимова и А.С.Поспелова, часть 2, М.2002.
3. А. Кривелёв. Основы компьютерной математики с использованием системы MatLab. М, 2005.