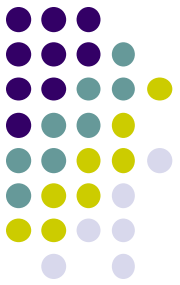


# Компьютерный практикум по математическому анализу в

## среде Matlab Практическое занятие 5

<http://serjmak.com/2students/matlabma/seminar5>  
<http://serjmak.com/2students/matlabma/seminar5.ppt>



### Темы

Файл-функции. Подфункции. Решение произвольных уравнений. Исследование локальных экстремумов и других характерных точек графиков функций одной и нескольких переменных. Полиномы, вычисление всех корней полинома.

Теория:

[http://serjmak.com/2students/matlabma/1.%20Matlab7\\_Anufr.pdf](http://serjmak.com/2students/matlabma/1.%20Matlab7_Anufr.pdf)

[1] (стр. 219-234, 246-264)

# Matlab: краткая теория



Возможные функции для выполнения заданий:

Файл-функции создаются в m-файлах:

```
function f = myfun(x) % файл-функция с 1 входным аргументом  
f=2*sin(x^2);
```

После сохранения функции можно обратиться к ней, как к встроенной:

```
>> y=myfun(2)
```

Однако обращение к функции происходит по имени файла, в котором она сохранена, а не по имени функции!

`fplot('myfun' или @myfun, [0 5], '.*')` – построение графиков на основе файл-функций, `@myfun` используется, если функция `myfun` находится в том же файле `m` и не сохранена как отдельный файл.

```
function f = radius3(x, y, z) % файл-функция с 3 входными аргументами
```

```
function [hour, minute, second] = hms(sec) % файл-функция с 3 выходными  
аргументами; её вызов:
```

```
[a, b, c] = hms(1000000000);
```

`function myfun(a, b)` или `function [ ] = myfun(a, b)` - это файл-функции, не возвращающие никаких значений.

`help`, `lookfor` – выводят информацию по первой строке (H1-line) комментариев после объявления функции.

# Matlab: краткая теория



Возможные функции для выполнения заданий:

Файл-функции и подфункции могут быть в одном m-файле, но файл-функция при этом может быть только одна (её имя должно совпадать с именем файла), подфункций в том же файле может быть сколько угодно. Начало новой подфункции означает конец тела предыдущей. Переменные в функциях и подфункциях локальные.

`global A` – глобальная переменная. Объявлять её надо и в теле основной функции, и в теле подфункции:

```
function myfun2; %лежит в myfun2.m
```

```
%основная функция
```

```
global A
```

```
A=2;
```

```
f1=f(A)
```

```
function z = f(x, y)
```

```
%подфункция
```

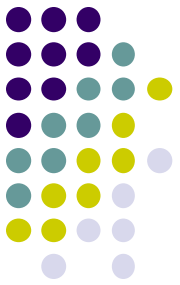
```
global A
```

```
z=x^3-2y^3-x*y+A;
```

Подфункции вызываются только с помощью `@`: `fplot(@f, [1 2])`

Подфункция доступна только внутри основной функции, не видна

# Matlab: краткая теория



Возможные функции для выполнения заданий:

`x=fzero('имя_функции', x0 или [0 4])` – имя файл-функции и начальное приближение к корню (какое-то значение) либо интервал, на котором

этот корень ищется; `fzero` приближённо вычисляет корень уравнения на некотором интервале или ближайший к заданному начальному приближению корень. Если указывается интервал, то на границах этого интервала функция должна принимать значения разных знаков, иначе будет ошибка.

Обращение с двумя параметрами: `[x, f] = fzero('sin', [-1 1]);`  
`fzero` вычисляет только те корни, в которых функция меняет знак, а не касается оси  $x$ . Поэтому, скажем, для  $x^2=0$  `fzero` корень не найдёт.

Функции можно задать с помощью `inline` и анонимно:

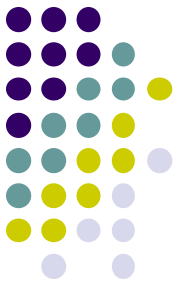
```
func1 = inline('sin(x)-x.^2.*cos(x)');
```

```
func2 = @(x) sin(x)-x.^2.*cos(x);
```

Ещё один тип задания `fzero`: `[x, f, flag] = fzero(func1, 0.1);` -

положительное значение `flag` свидетельствует об успешном вычислении, отрицательное – о том, что что-то пошло не так. Это можно использовать в алгоритмах.

# Matlab: краткая теория



Возможные функции для выполнения заданий:

Полином задаётся вектором его коэффициентов:

$x^3+3x^2+3 \rightarrow p = [1 \ 3 \ 0 \ 3]$  (считая нулевые коэффициенты).

`polyval(p, 1)` считает значение полинома от аргумента (в данном случае 1). Аргумент может быть матрицей или вектором.

`roots(p)` вычисляет корни полинома (аргументом служит вектор коэффициентов).

`fminbnd(@funcname, -10, 10)` – нахождение локального минимума функции одной переменной на указанном интервале от -10 до 10.

В остальном её вызов аналогичен `fzero`.

`fminsearch(@funcname, -10)` - нахождение локального минимума функции нескольких переменных (в том числе и одной) с указанным начальным приближением -10. В остальном её вызов аналогичен `fzero` или `fminbnd`. В случае функции нескольких переменных начальным приближением является вектор:

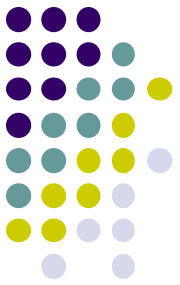
`fminsearch(@funcname, [1, 2])` (для функции двух переменных).

## Matlab: задание



- 1) Постройте график функции  $f = \exp(-x) \cdot \sqrt{(x^2+1)/(x^4+0.1)}$  на интервале  $[0, 4]$  с шагом 0.1, сделав из неё файл-функцию myfun1.
- 2) Постройте графики функции из п.1 на одних осях, используя plot и fplot. Добавьте в myfun1 комментарии на H1-line и убедитесь, что help и lookfor выдают нужную информацию по myfun1.
- 3) Напишите файл-функцию, вычисляющую сумму всех элементов вектора с нечётными индексами.
- 4) Создайте файл-функцию myfun2, содержащую функцию для решения уравнения  $\sin(x) - x^2 \cdot \cos(x) = 0$  на интервале  $[-5; 5]$ . Перед использованием fzero постройте график с помощью fplot с сеткой, и найдите с помощью fzero все 4 корня уравнения, задавая начальные приближения x0 в соответствии с графиком.
- 5) Найдите все корни полинома  $x^9+3x^4+x^3-10x^2-x+1024$  и вычислите значение полинома от получившегося вектора его корней.
- 6) Найдите все локальные минимумы функции  $y = e^{(x^2)} + \sin 3\pi x$ , задав её с помощью inline. Постройте график функции с помощью fplot, чтобы знать интервалы и начальные точки поиска.
- 7) Найдите все локальные минимумы для функции  $y = x^2+1$ . (задайте анонимную функцию, fplot).

## Matlab: задание



- 8) Минимизируйте функцию двух переменных  $f(x, y) = \sin \pi x * \sin \pi y$ , для этого постройте линии уровня этой функции с помощью `meshgrid` и `contour` на участке  $[0, 2]$  с шагом 0.01 (см. [1]) с параметрами -0.96, -0.9, -0.8, -0.5, -0.1, ..., 0.1, 0.5, 0.8, 0.9, 0.96, подписав ими сами линии (`clabel`). Затем создайте файл-функцию, содержащую функцию и вектор  $[x_0, y_0]$  в качестве входного параметра. После этого, с помощью `fminsearch` и различных векторов приближений найдите все 4 локальных экстремума функции. Постройте освещённую поверхность функции и проанализируйте экстремумы, повертев поверхность.
- 9) Результат (файл \*.m) - мне на почту.