Файл-функции и скрипты

Основы MATLAB

Юдинцев В. В.

Кафедра теоретической механики

23 февраля 2026 г.

САМАРСКИЙ УНИВЕРСИТЕТ
SAMARA UNIVERSITY

Содержание

Функции пользователя

Способы задания функции

- inline функции;
- анонимные функции;
- файл-функции.

inline-функция

```
>> f1 = inline('x1^2+x2^2','x1','x2');
>> f1(4,2)
ans =
    20
```

Переменные из рабочей области недоступны.

Анонимная функция

```
>> f = @(x1,x2) x1^2+x2^2;
>> f(4,2)
ans =
    20
```

Переменные из рабочей области доступны, но рассматриваются как константы:

```
>> a=2;
>> f = @(x1) a*x1;
>> f(1)
ans =
    2
>> a=1; f(1)
ans =
    2
```

Файлы-функции и скрипты

Типы файлов

- `*.m` (текстовые) содержат тексты программ, определения функций.
- `*.mat` (бинарные) содержат значения переменных.
- `*.mex` (бинарные). MEX-файлы – динамически подключаемые библиотеки

Скрипты

- Любую последовательность команд в MATLAB можно оформить в виде m файла.
- По умолчанию все переменные, объявленные внутри файл-скрипта, являются глобальными.

Функции

- Файл-функция содержит определение одной или нескольких функций.
- По-умолчанию все переменные, объявленные внутри файл-функции, являются локальными.
- Файл-функция является самостоятельным программным модулем, который связан с другими модулями и головной программой через входные и выходные параметры.

Создание m-файлов

- При создании файл-функций и файл-скриптов следует избегать перекрытия имён других функций.
- Для проверки имени можно использовать функцию `exist`:

```
>> exist atan
ans =
     5
```

- Если имя не занято, то функция `exist` возвращает 0 (5 – если имя занято встроенной функцией).

Файл-скрипты

Файл-скрипт

- Файл-скрипт не имеет входных и выходных аргументов.
- Работает с данными из рабочей области.
- Все переменные, объявленные в файл-скрипте, являются глобальными.
- В процессе выполнения не компилируются.
- Представляют собой зафиксированную в виде файла последовательность операций.

Структура файл-скрипта

Файл `fscript.m`

```
1  % Описание кода,
2  % которое можно увидеть
3  % напечатав help fscript в окне команд
4  x=1:0.1:10;
5  y=sin(x);
```

Выполнение файл-скрипта

- Вызов из командной строки (command window)
- Запуск из редактора при помощи сочетания Ctrl Enter
- Файл-скрипт можно для удобства разделить на ячейки (секции) при помощи удвоенного знака %%

```
%% секция 1
код ...
%% секция 2
код ...
```

- Код в каждой секции может быть выполнен при помощи Ctrl Enter

Файл-функции

Объявление функции

File New Function

```
1   % комментарии к функции,
2   % которые будут выводится
3   % по команде
4   % help func_name
5   function [out1,out2] = func_name(in1,in2)
6    %
7    % тело функции
8    %
9    out1 = ...
10   out2 = ...
11
```

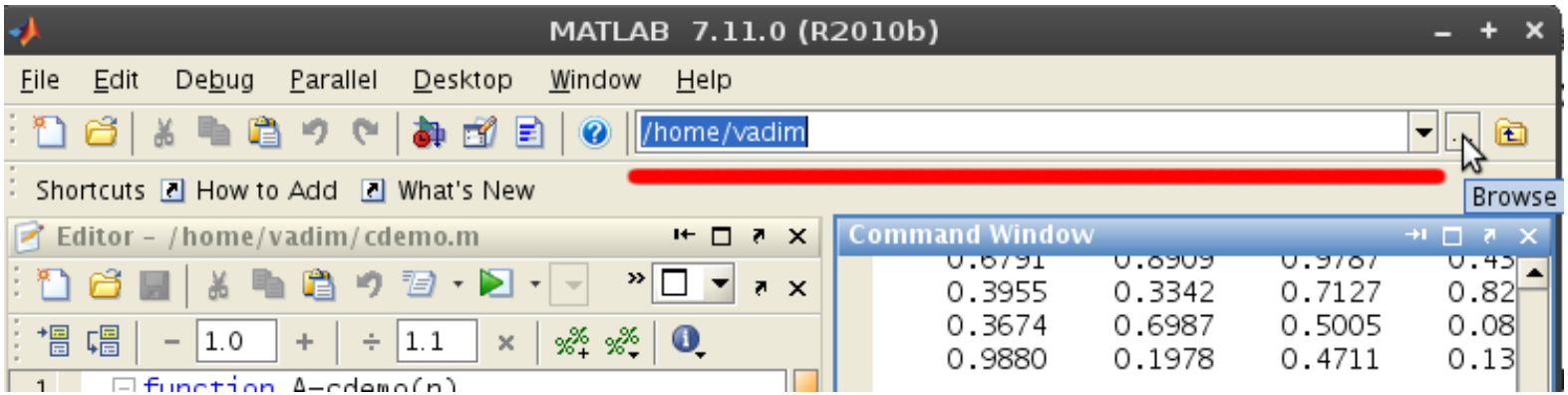
Глобальные переменные в функциях

- По умолчанию все переменные внутри функции являются локальными.
- Для того, чтобы несколько функций использовали одну переменную, её необходимо объявить глобальной.

```
1  function res = func_name(in1,in2)
2   global G;
3   res=G*in1+in2;
```

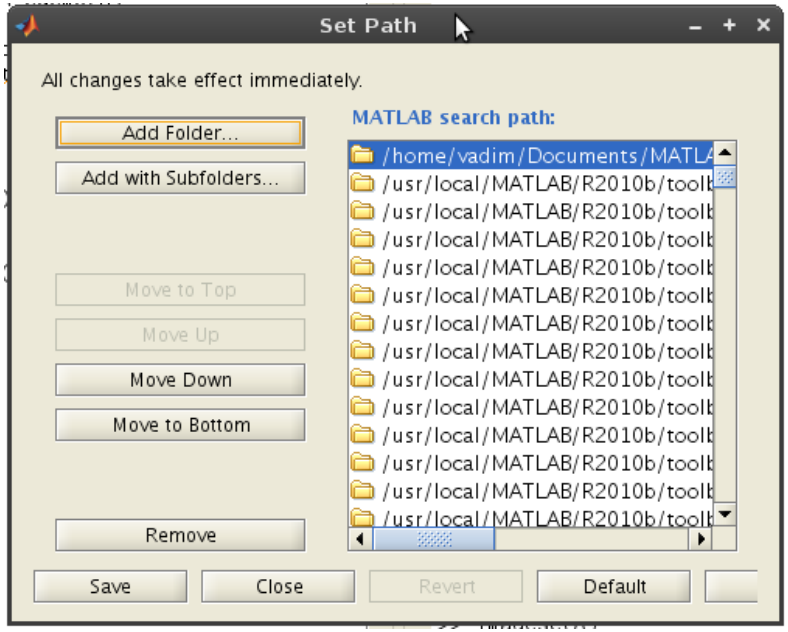
Выполнение функций и файл-скриптов

Имя файла и имя объявленной в нем функции предпочтительно делать одинаковыми. Каталог, в котором содержатся вызываемые функции, должен быть текущим или добавлен в пути поиска.



Выполнение функций и файл-скриптов

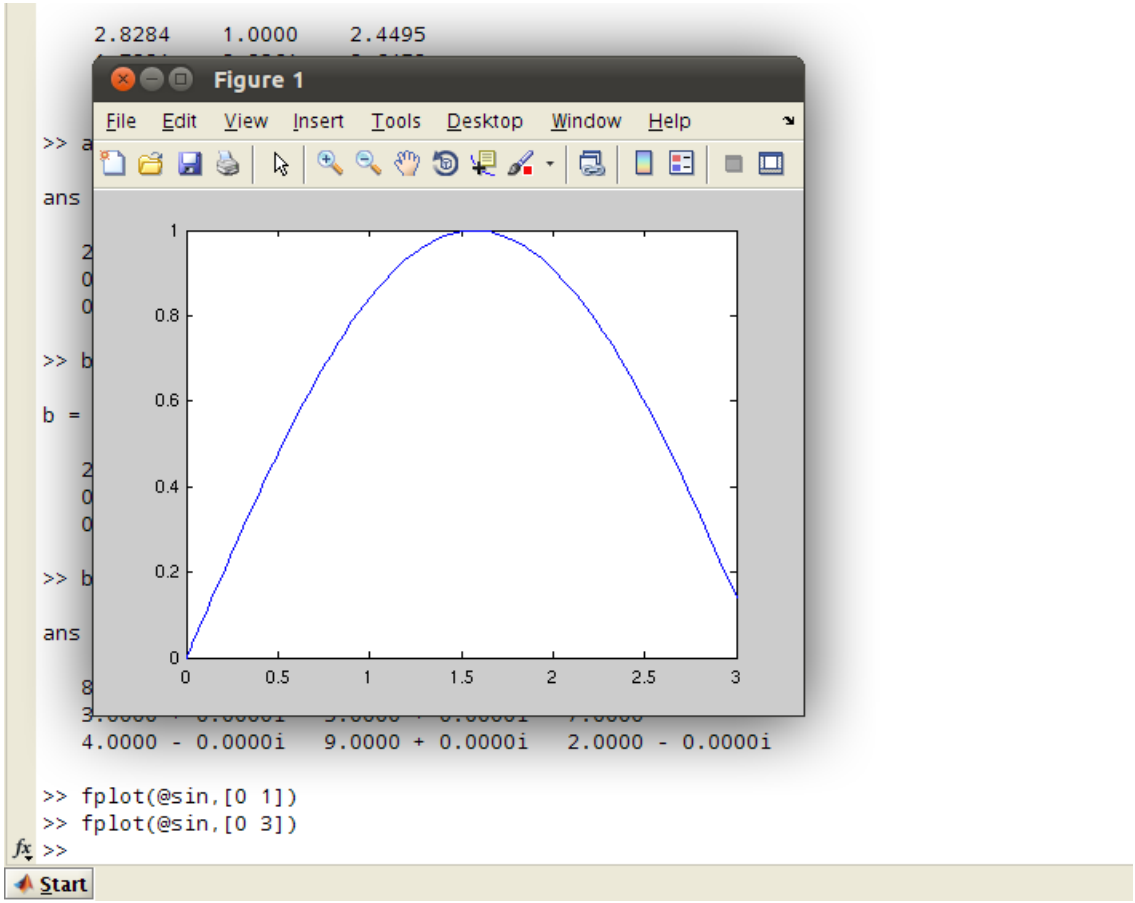
File Set Path



Оператор return

- Функция прекращает работу после выполнения последнего оператора.
- Принудительно завершить функцию можно оператором `return`.

Быстрое построение графика функции



Функция `fplot`

```
1 fplot('myfun',[0 4])
2 fplot(@myfun,[0 4])
```

первый аргумент: имя или ссылка на функцию, второй аргумент: диапазон изменения аргумента функции для построения графика

Внутренние функции

- Файл-функция вместе с определением основной функции может содержать определения вспомогательных функций, доступных к вызову только из основной функции.

```
function f = myfun(x)
  f1=infun(x);
  f=f1+cos(x);
% Внутренняя функция
function f = infun(x)
  a=3;
  f=sin(x*3);
```

- Переменные, используемые в подфункциях локальные.

Вложенные функции

- Вложенная функции определяется в теле основной функции. Файл `myfun.m`:

```
1 function f = myfun(x)
2   f1=infun(x);
3   f=f1+cos(x);
4   function f = infun(x)
5     f=sin(x);
6   end
7 end
```

- Из вложенной функции `infun` доступны локальные переменные всех функций верхнего уровня и наоборот.

Функции с переменным количеством аргументов

- `varargin` – список ячеек с параметрами функции;
- `length(varargin)` – количество переданных аргумента;
- `varargin{1}` – первый аргумент;
- `varargin{2}` – второй аргумент.

```
1 function res = fun(varargin)
2   if length(varargin)<2
3     error('Недостаточное количество аргументов');
4   end
5   ...
```

Функции с переменным количеством аргументов

Аргумент `varargin` может быть указан после перечня обязательных аргументов

```
1 function res = fun(a1,a2,varargin)
2   if length(varargin)<4
3     error('Недостаточное количество аргументов');
4   end
5   ...
```

Функция в качестве аргумента

Использование функции `feval`

Передача имени функции строкой:

```
>> x = 1;
>> feval('sin',x)
ans =
    0.8415
```

Передача ссылки на функцию (это работает быстрее):

```
>> feval(@sin,x)
ans =
    0.8415
```