



*Российский государственный университет  
нефти и газа им. И.М. Губкина*

*Кафедра Информатики*

*Дисциплина: Программные комплексы  
общего назначения*

*Преподаватель:*

**К.Т.Н., ДОЦЕНТ**

**Коротаев**

**Александр Фёдорович**



# Локальные и глобальные переменные

Каждая m-функция располагает собственным пространством переменных (**локальные переменные**). Они изолированы от рабочего пространства других функций и рабочей области системы MatLab.

Поэтому переменные с одинаковыми именами в разных функциях никак не связаны друг с другом.

Однако есть возможность объявить переменную **глобальной**. Любое присваивание значения этой переменной распространяется на все функции, где она объявлена глобальной.

Например, описание  
**global ALPHA BETA**

Объявляет переменные **ALPHA** и **BETA** глобальными

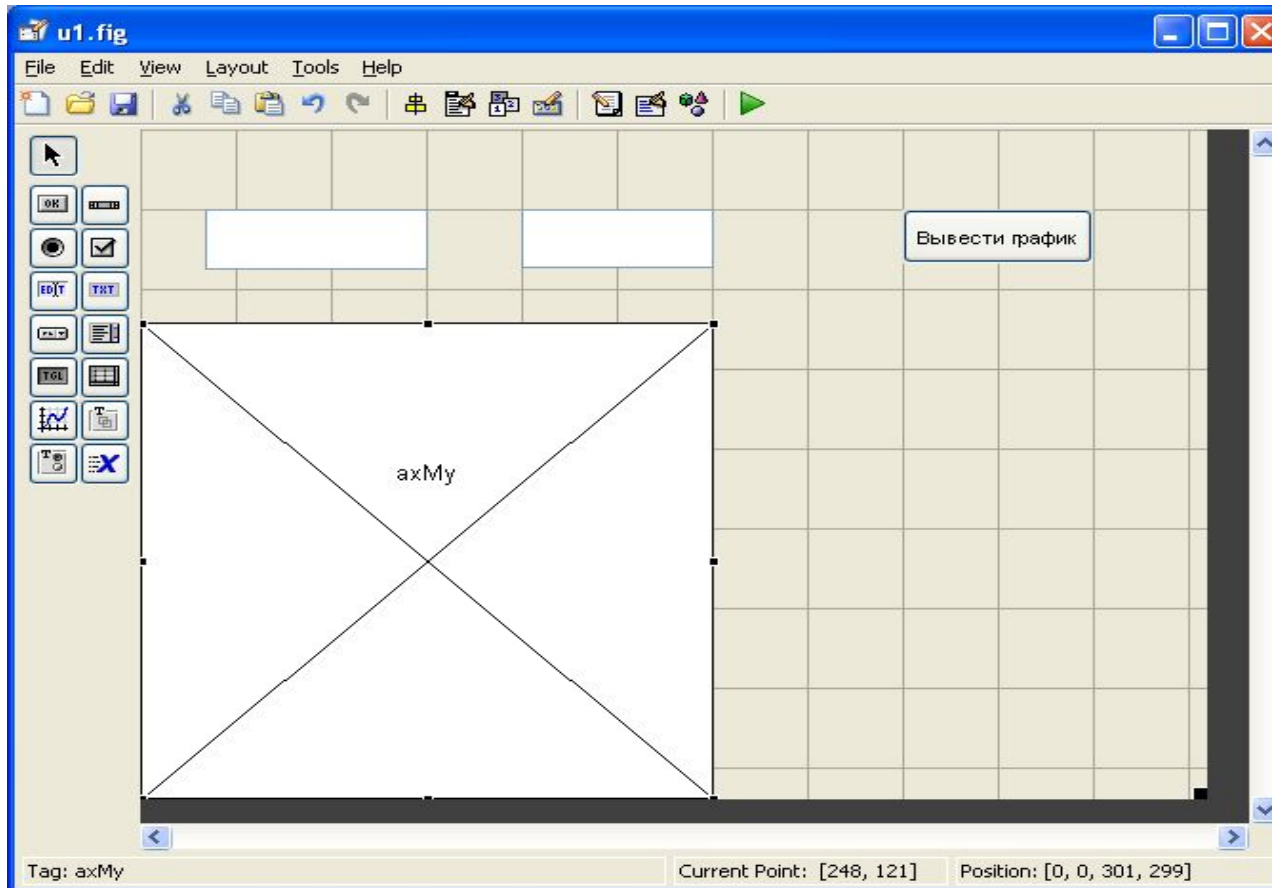


# Локальные и глобальные переменные

Для работы с **глобальными переменными** необходимо:

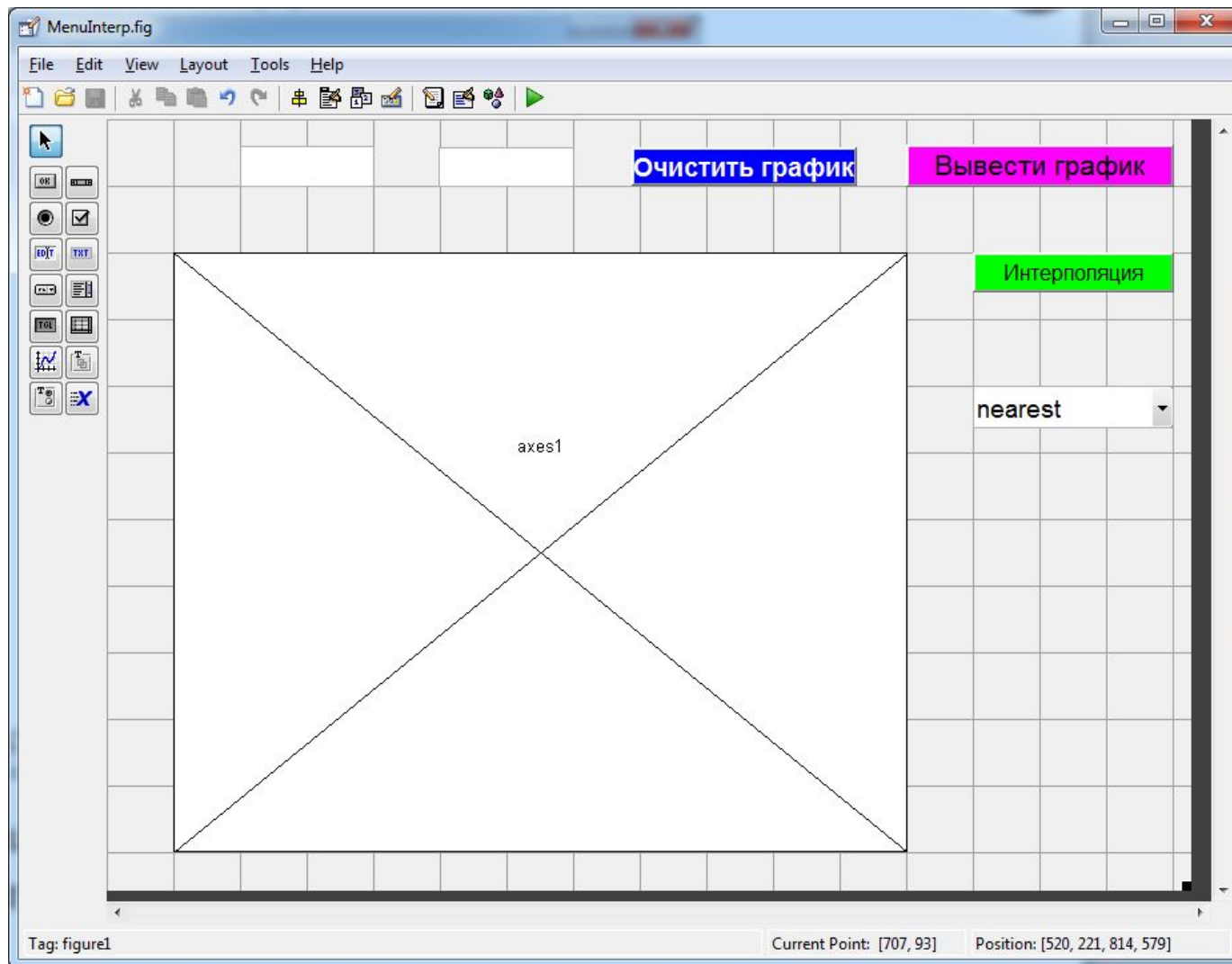
- объявить переменную как глобальную в **каждой m-функции**, которой необходима эта переменная
- объявить переменную как глобальную из **командной строки**, если необходимо сделать её **глобальной в рабочей области**
- в **каждой m-функции** использовать команду **global** перед первым появлением переменной (желательно в начале m-файла)

Пусть имеется интерфейс для построения точечного графика функции



```
function pushbutton1_Callback(hObject, eventdata, handles)  
f=inline(get(handles.edit1,'String'));  
a=str2num(get(handles.edit2,'String'));  
X=linspace(a(1),a(2),11);  
Y=f(X)  
plot(X,Y,'*r');
```

И мы хотим дополнить интерфейс до следующего:



Для этого нам надо создать соответствующие графические элементы и настроить их

Теперь нужно описать действия по нажатию кнопки **Интерполяция**

```
function pushbutton2_Callback(hObject, eventdata, handles)
```

```
% hObject   handle to pushbutton2 (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles   structure with handles and user data (see GUIDATA)
```

```
global a f X Y
```

```
temp1=get(handles.menu1,'String')
```

```
temp2=get(handles.menu1,'value')
```

```
met=temp1{temp2}
```

```
xi=linspace(a(1),a(2),101);
```

```
yi=interp1(X,Y,xi,met);
```

```
plot(xi,yi,'g')
```



# Интеграция MatLab и Excel

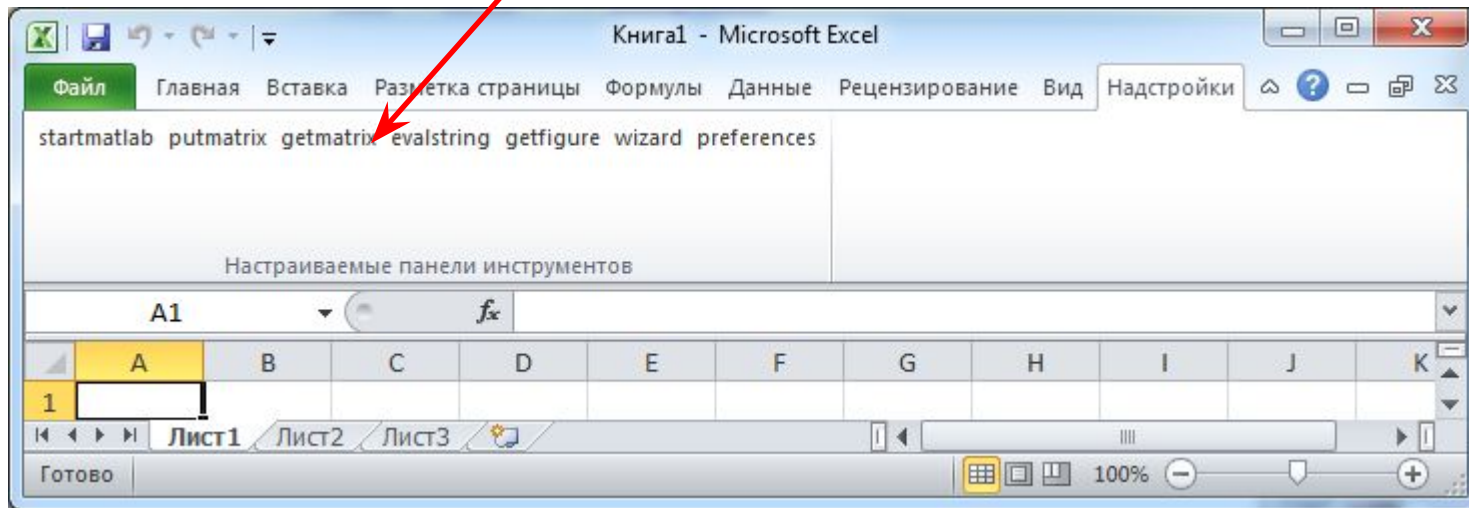
Позволяет пользователю **Excel** обращаться к многочисленным функциям **MatLab** с целью обработки данных, реализации вычислений, визуализации результатов. Такое расширение возможностей Excel требует подключения надстройки **ExcelLink**:

- В диалоговом окне меню **Файл/Параметры/Надстройки...** укажите путь к файлу **exclink.xla** (В подкаталоге **toolbox\exclink** основного каталога **MatLab**)
- В списке надстроек диалогового окна появится строка **Spreadsheet Link EX 3.1.5 for use with MatLab**
- Нужно установить флажок в этой строке для подключения надстройки



# Интеграция MatLab и Excel

Теперь в **Excel** присутствует вкладка с панелью инструментов для связи с **MatLab**

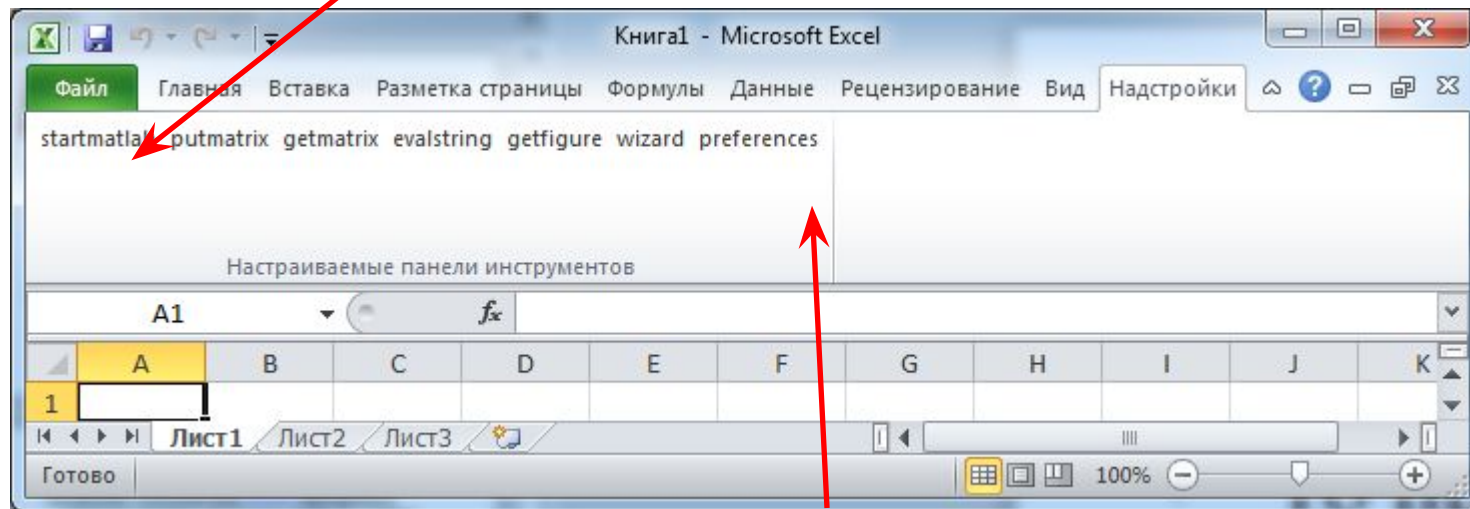






# Интеграция MatLab и Excel

Если надстройка **Spreadsheet Link EX** активна, то при запуске **Excel** автоматически запустится и **MatLab**. В противном случае нажимаем кнопку **startmatlab**



Дополнительные установки можно сделать с помощью кнопки **preferences**



# Обмен данными между MatLab и Excel

Для переноса матрицы из **Excel** в **MatLab**

нужно выделить содержащие её ячейки и

нажать кнопку **putmatrix**

Появляется диалоговое окно

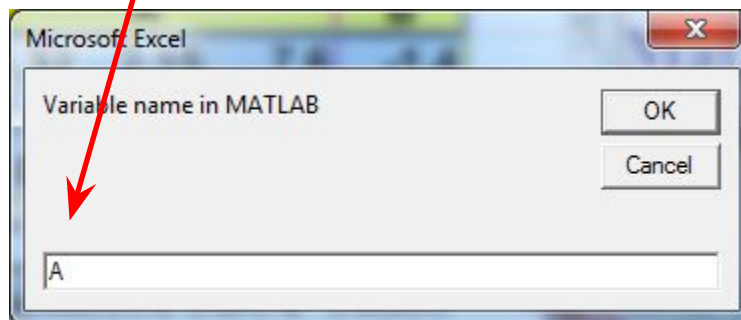
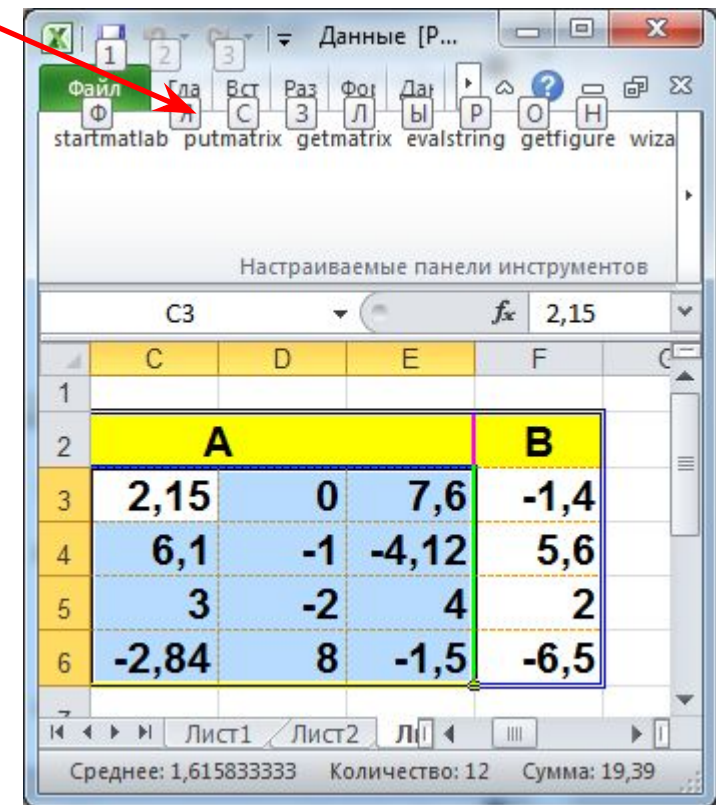
**Excel**, в которую следует

ввести **имя переменной** для

рабочей среды **MatLab**, куда

будут экспортированы данные

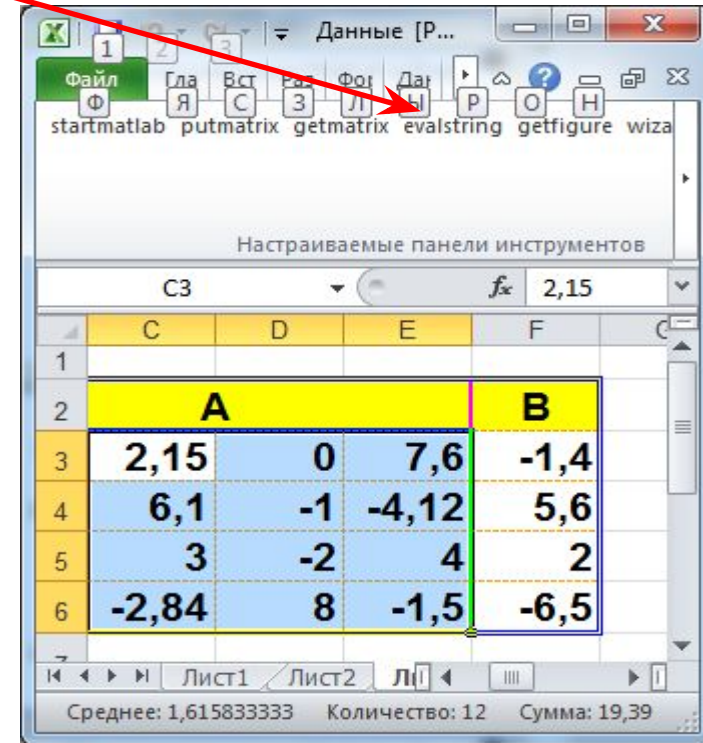
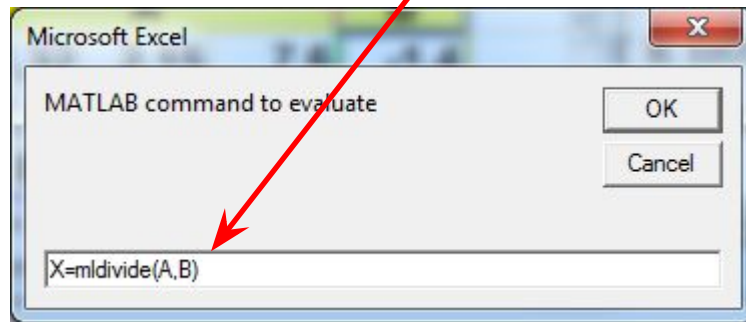
из выделенных ячеек



# Обмен данными между MatLab и Excel



Нажатие на кнопку **evalstring** приводит к появлению диалогового окна, в которое можно ввести команду **MatLab**.  
Например, **mldivide**

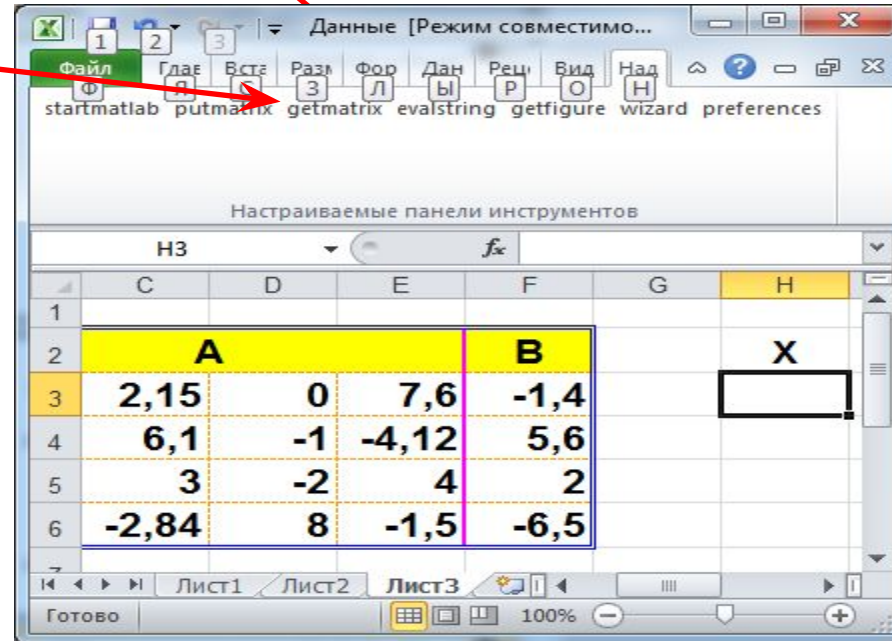


Результат будет такой же, как и при выполнении команды непосредственно в среде MatLab.

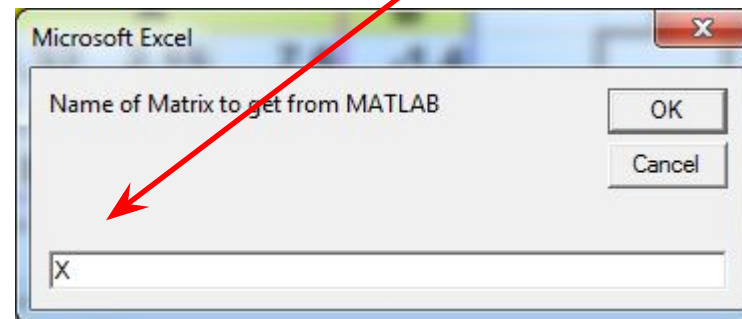
# Обмен данными между MatLab и Excel



Для импорта результатов обратно в **Excel** нужно выделить **начальную ячейку** и нажать кнопку **getmatrix**



В диалоговое окно требуется ввести **имя переменной**, импортируемой в **Excel**

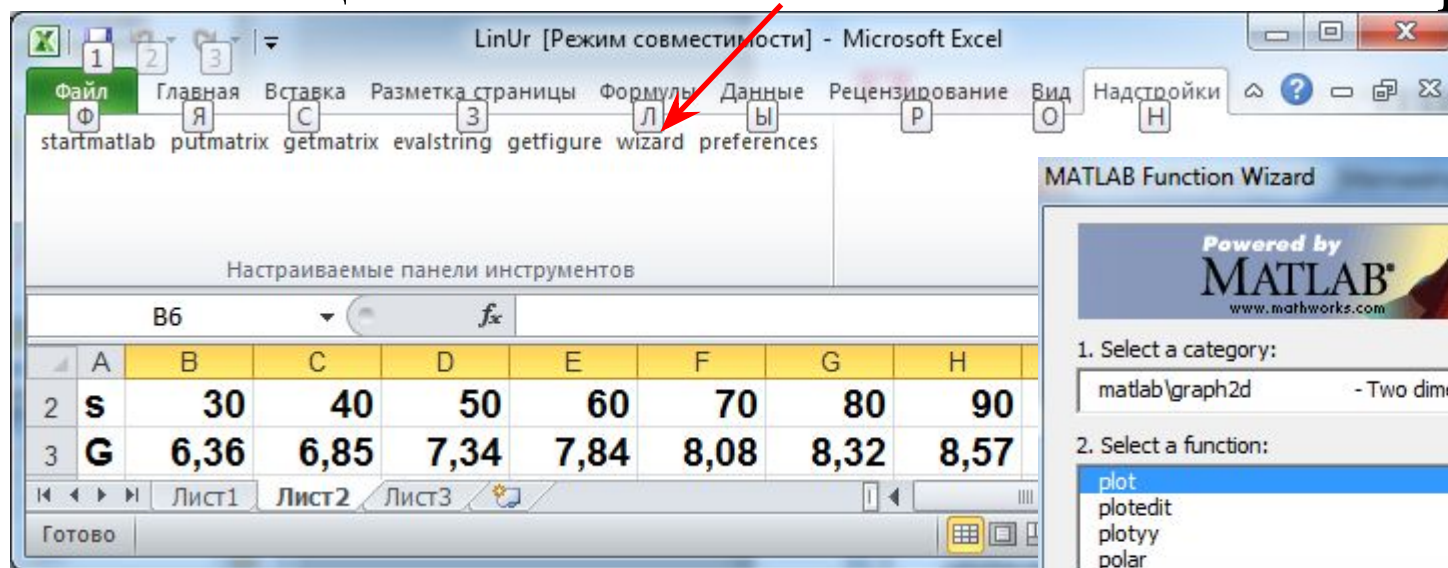




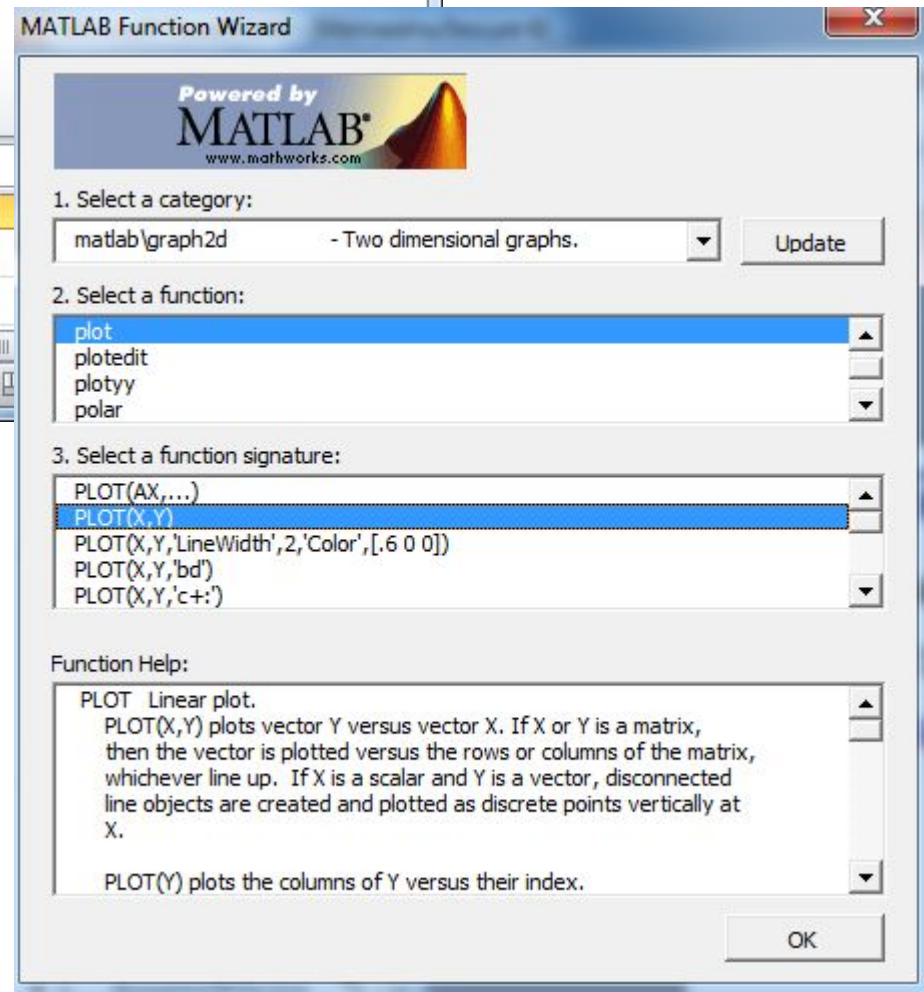
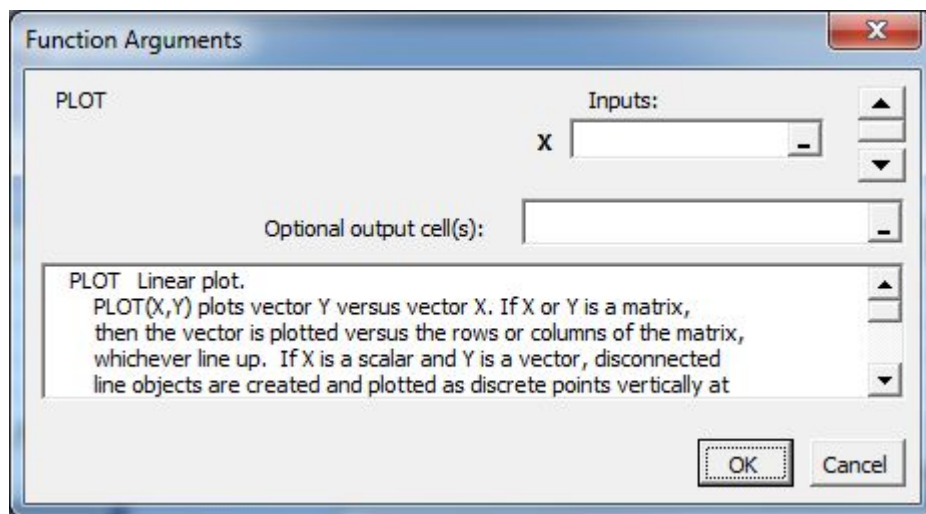
# Обмен данными между MatLab и Excel



С помощью кнопки **vizard** можно вызвать функцию **MatLab**



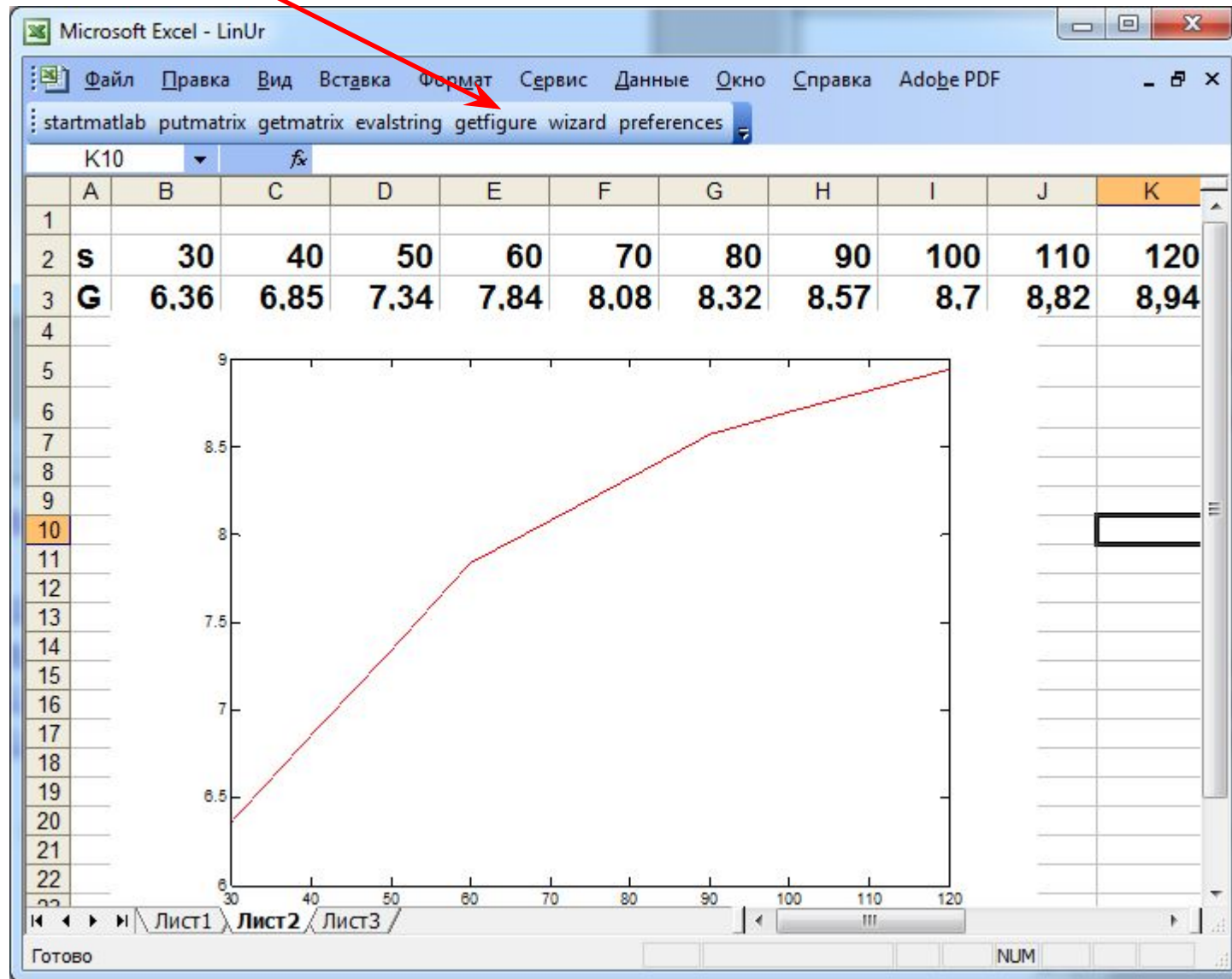
Пример: функция **plot**



# Обмен данными между MatLab и Excel

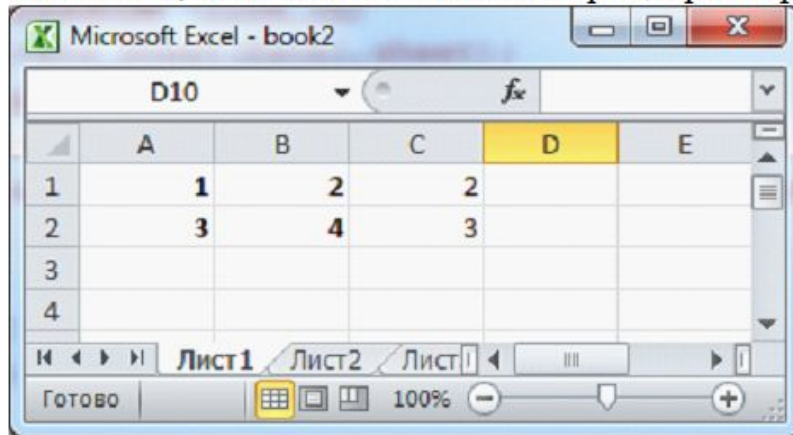


Кнопка **getfigure** импортирует из **MatLab** фигуру



# Примеры тестовых вопросов

Укажите все варианты, с помощью которых из файла, созданного в Excel2010, числовые значения матрицы размером 2\*3 передаются в Matlab?



Текст вопроса

Варианты ответов

- ☒ A = xlsread( 'book2. xlsx')
- ☐ A = xlsread( 'book2. xlsx', '1', 'A1: C2')
- ☒ A = xlsread( 'book2. xlsx', 'A1: C2')
- ☒ A = xlsread( 'book2. xlsx', 'Лист1', 'A: C')
- ☐ A = xlsread( 'book2. xls', 'Лист1', 'A1: C2')

# Примеры тестовых вопросов

На сколько байт сдвинется указатель файла после выполнения фрагмента программы?

```
a=[ 1.1  8  5.1  8.1 ], b=[ 1.1  2.1  1 ]
```

```
fid1 = fopen( 'data. txt', 'w' );
```

```
fwrite( fid1, a, 'float' );
```

Эталон ответа

16

Какое действие реализует функция `interp1( xi, y, x, 'linear' )`?

Варианты ответов

- ☒ Кусочно - линейная интерполяция по данным  $x$  и  $y$  в точке  $x_i$
- ☐ Так нельзя записывать функцию
- ☐ Кусочно - линейная интерполяция по данным  $x_i$  и  $y$  в точке  $x$
- ☐ Ступенчатая интерполяция по данным  $x_i$  и  $y$  в точке  $x$