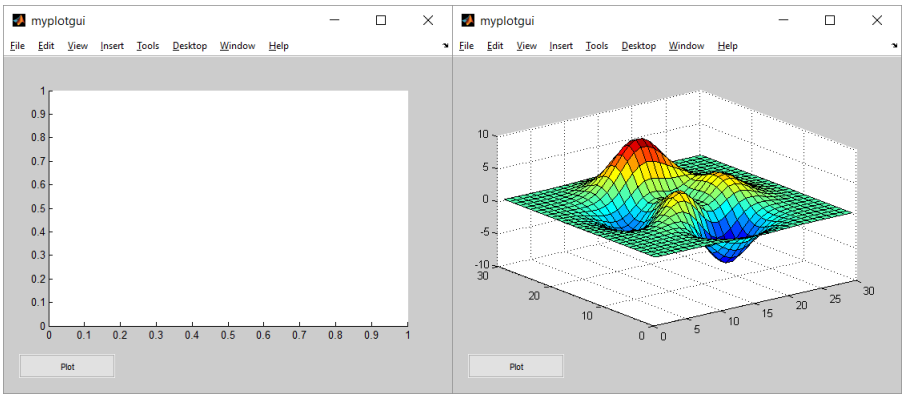
GUI программным способом

Пример простого GUI

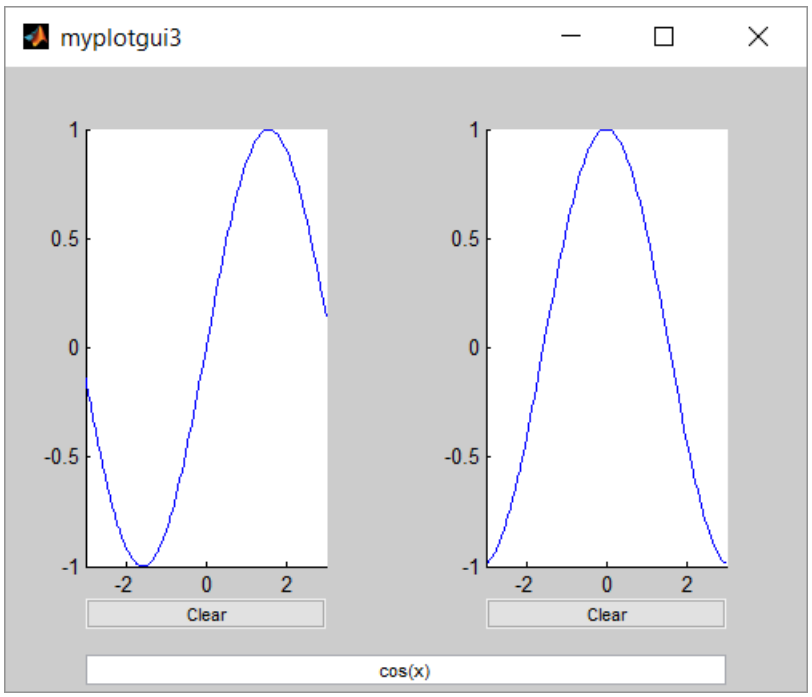
```
function myplotgui
% создание графического окна с заголовком myplotgui и без надписи Figure
1
hF = figure('Name', 'myplotgui', 'NumberTitle', 'off');
% создание осей
hA = axes('Position', [0.1 0.2 0.8 0.7]);
% создание кнопки и задание свойству Callback значения указателя на
подфункцию
hBtnPlot = uicontrol('Style', 'pushbutton', ...
     'Position', [20 20 120 30],...
    'String', 'Plot',...
    'Callback', @BtnPlotCallback);

function BtnPlotCallback(src,evt)
% подфункция обработки события Callback кнопки Plot
surf(peaks(30))
```



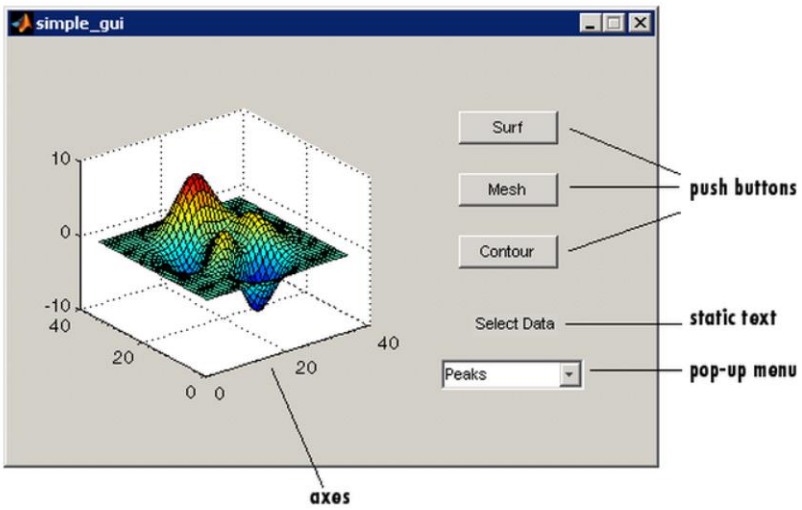
Пример построение графиков в двух окнах GUI

```
function myplotgui515
% создаем графическое окно с тегом win
hF = figure('Name', 'myplotgui3', 'NumberTitle','off',...
    'MenuBar', 'none', 'Units', 'characters',...
    'Position', [10 10 100 30], 'Tag', 'win');
% создаем оси с тегом axLeft
axes('Position', [0.1 0.2 0.3 0.7], 'Tag', 'axLeft',...
    'NextPlot', 'replacechildren');
% создаем оси с тегом axRight
axes('Position', [0.6 0.2 0.3 0.7], 'Tag', 'axRight',...
    'NextPlot', 'replacechildren');
% создаем кнопку с тегом btnLeft
uicontrol('Style', 'pushbutton', 'Units', 'normalized',...
    'Position', [0.1 0.1 0.3 0.05],...
    'String', 'Plot', 'Callback', @BtnLeftCallback,...
    'Tag', 'btnLeft');
% создаем кнопку с тегом btnRight
uicontrol('Style', 'pushbutton','Units', 'normalized',...
    'Position', [0.6  0.1 0.3 0.05],...
    'String', 'Plot', 'Callback', @BtnRightCallback,...
    'Tag', 'btnRight');
% создаем область ввода текста с тегом edtFun
```

```
uicontrol('Style', 'edit', 'Units', 'normalized',...
    'Position', [0.1  0.01 0.8 0.05],...
    'BackgroundColor', 'w',...
    'Tag', 'edtFun');
% скрываем указатель на окно приложения
set(hF, 'HandleVisibility', 'callback');

function BtnLeftCallback(src, evt)
% подфункция обработки события нажатия на левую кнопку
% записываем в структуру handles указатели на объекты приложения
handles = guihandles(src);
% сейчас в поле axLeft структуры handles находится указатель на левые оси
% делаем их текущими
axes(handles.axLeft)
% Проверяем, совпадает ли надпись на кнопке с Plot
if isequal(get(src, 'String'), 'Plot')
    % надпись на кнопке Plot
    % берем текст из строки ввода (выражение для функции)
    str = get(handles.edtFun, 'String');
    % строим график на левых осях (они текущие)
    fplot(str, [-3 3])
    % изменяем надпись на кнопке на Clear
    set(src, 'String', 'Clear')
else
    % очищаем левые оси (они текущие)
    cla
    % изменяем надпись на кнопке на Plot
    set(src, 'String', 'Plot')
end

function BtnRightCallback(src, evt)
% подфункция обработки события нажатия на правую кнопку
% записываем в структуру handles указатели на объекты приложения
handles = guihandles(src);
% сейчас в поле axRight структуры handles находится указатель на правые оси
% делаем их текущими
axes(handles.axRight)
% Проверяем, совпадает ли надпись на кнопке с Plot
if isequal(get(src, 'String'), 'Plot')
    % надпись на кнопке Plot
    % берем текст из строки ввода (выражение для функции)
    str = get(handles.edtFun, 'String');
    % строим график на правых осях (они текущие)
    fplot(str, [-3 3])
    % изменяем надпись на кнопке на Clear
    set(src, 'String', 'Clear')
else
    % очищаем правые оси (они текущие)
    cla
    % изменяем надпись на кнопке на Plot
    set(src, 'String', 'Plot')
end
```



Пример GUI с тремя кнопками и выпадающим меню



Последовательность действий

1. Создайте файл-функцию

```
function mysimple_gui
```

2. Создание окна приложения

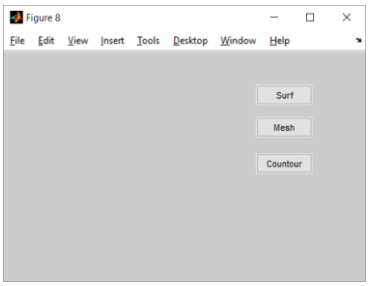
```
f = figure('Visible','on','Position',[360,500,450,285]);
```

Свойство Position представляет собой четырехэлементный вектор, который определяет расположение графического интерфейса на экране и его размер: [расстояние от левого края, расстояние от нижнего края, ширина, высота]. Единицы по умолчанию - пиксели.

3. Создание элементов приложения . Три кнопки

```
hsurf     = uicontrol('Style','pushbutton',...
              'String','Surf','Position',[315,220,70,25]);
```

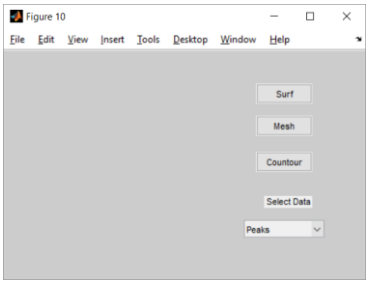
```
hmesh    = uicontrol('Style','pushbutton',...
             'String','Mesh','Position',[315,180,70,25]);

hcontour = uicontrol('Style','pushbutton',...
             'String','Countour','Position',[315,135,70,25]);
```



Каждый оператор использует серию пар свойство- значение uicontrol для определения кнопки:
Свойство Style указывает, что uicontrol является кнопкой.
Свойство String задает метку для каждой кнопки: Surf, Mesh и Contour.
Свойство Position определяет расположение каждой кнопки в GUI и ее размер.

4. Создание элементов приложения. Меню и текст

```
hpopup = uicontrol('Style','popupmenu',...
           'String',{'Peaks','Membrane','Sinc'},...
           'Position',[300,50,100,25]);
htext  = uicontrol('Style','text','String','Select Data',...
           'Position',[325,90,60,15]);
```



В свойстве String свойства pop-menu используется массив ячеек, чтобы указать три элемента во всплывающем меню: Peaks, Membrane и Sinc.

5. Добавление осей

```
ha = axes('Units','pixels','Position',[50,60,200,185]);
```

Свойство Units определяет пиксели так, чтобы оси имели те же единицы, что и остальные компоненты.

6. Выравнивание компонентов относительно их центра (кроме осей)

```
align([hsurf,hmesh,hcontour,htext,hpopup],'Center','No');
```

7. Сделать объект видимым

```
set(f,'Visible','on')
```

8. Программа для меню

```
   function popup_menu_Callback(source,eventdata)
      % Determine the selected data set.
      str = get(source, 'String');
      val = get(source,'Value');
      % Set current data to the selected data set.
      switch str{val};
      case 'Peaks' % User selects Peaks.
         current_data = peaks_data;
      case 'Membrane' % User selects Membrane.
         current_data = membrane_data;
      case 'Sinc' % User selects Sinc.
         current_data = sinc_data;
      end
   end
```

9. Программирование кнопок

```
function surfbutton_Callback(source,eventdata)
% Display surf plot of the currently selected data.
     surf(current_data);
end

function meshbutton_Callback(source,eventdata)
% Display mesh plot of the currently selected data.
     mesh(current_data);
end

function contourbutton_Callback(source,eventdata)
% Display contour plot of the currently selected data.
     contour(current_data);
end
```

10. Программирование обратного вызова

```
hsurf = uicontrol('Style','pushbutton','String','Surf',...
        'Position',[315,220,70,25],...
        'Callback',{@surfbutton_Callback});

'Callback',{@meshbutton_Callback}

'Callback',{@contourbutton_Callback}

'Callback',{@popup_menu_Callback}
```

Результат

```
function simple_gui56
f = figure('Visible','off','Position',[360,500,450,285]);

% Construct the components.
hsurf    = uicontrol('Style','pushbutton',...
            'String','Surf','Position',[315,220,70,25],...
            'Callback',{@surfbutton_Callback});
```

```
hmesh    = uicontrol('Style','pushbutton',...
             'String','Mesh','Position',[315,180,70,25],...
             'Callback',{@meshbutton_Callback});
hcontour = uicontrol('Style','pushbutton',...
             'String','Countour','Position',[315,135,70,25],...
             'Callback',{@contourbutton_Callback});
htext  = uicontrol('Style','text','String','Select Data',...
           'Position',[325,90,60,15]);
hpopup = uicontrol('Style','popupmenu',...
           'String',{'Peaks','Membrane','Sinc'},...
           'Position',[300,50,100,25],...
           'Callback',{@popup_menu_Callback});
ha = axes('Units','pixels','Position',[50,60,200,185]);
align([hsurf,hmesh,hcontour,htext,hpopup],'Center','None');

% Initialize the GUI.
% Change units to normalized so components resize automatically.
set([f,hsurf,hmesh,hcontour,htext,hpopup],'Units','normalized');

% Generate the data to plot.
peaks_data = peaks(35);
membrane_data = membrane;
[x,y] = meshgrid(-8:.5:8);
r = sqrt(x.^2+y.^2) + eps;
sinc_data = sin(r)./r;

% Create a plot in the axes.
current_data = peaks_data;
surf(current_data);

% Assign the GUI a name to appear in the window title.
set(f,'Name','Simple GUI')

% Move the GUI to the center of the screen.
movegui(f,'center')

% Make the GUI visible.
set(f,'Visible','on');

%  Pop-up menu callback. Read the pop-up menu Value property to
%  determine which item is currently displayed and make it the
%  current data. This callback automatically has access to
%  current_data because this function is nested at a lower level.
   function popup_menu_Callback(source,eventdata)
      % Determine the selected data set.
      str = get(source, 'String');
      val = get(source,'Value');
      % Set current data to the selected data set.
      switch str{val};
      case 'Peaks' % User selects Peaks.
         current_data = peaks_data;
      case 'Membrane' % User selects Membrane.
         current_data = membrane_data;
      case 'Sinc' % User selects Sinc.
         current_data = sinc_data;
      end
   end

  % Push button callbacks. Each callback plots current_data in the
  % specified plot type.

  function surfbutton_Callback(source,eventdata)
  % Display surf plot of the currently selected data.
       surf(current_data);
  end

  function meshbutton_Callback(source,eventdata)
  % Display mesh plot of the currently selected data.
       mesh(current_data);
  end

  function contourbutton_Callback(source,eventdata)
```

```
  % Display contour plot of the currently selected data.
       contour(current_data);
  end
end
```

Функции GUI

- uicontrol – управление пользовательским интерфейсом;
- uimenu – создание меню пользователя;
- ginput – графический ввод с помощью мыши (работа команды заканчивается нажатием клавиши Enter или устанавливается число обращений ginput(n));
- dragrect – создание и перетаскивание прямоугольника с помощью мыши;
- rbbox – растягивание прямоугольника мышью;
- selectmoveresize – выделение, копирование, перемещение и изменение размеров объекта;
- waitforbuttonpress – задержка выполнения программы до нажатия кнопки мыши;
- waitfor – блокировка исполнения и ожидание события;
- uiwait – прекращение выполнения программы до команды uiresume или закрытия окна;
- uiresume – возобновление работы программы;
- uisuspend – запрет интерактивного состояния графического окна;
- uirestore – восстановление интерактивного состояния;
- guide – создание интерфейса в интерактивном режиме;
- align - выравнивание объектов интерфейса;
- cbedit - редактирование объектов интерфейса в интерактивном режиме;
- menuedit – изменение меню;
- propedit – изменение свойств объекта;
- dialog – создание диалогового окна;
- axlimdlg – вызов окна для изменения координатных осей графического окна;
- errordlg – создание окна с сообщением об ошибке;
- helpdlg – создание справочного окна;
- inputdlg – создание окна для ввода;
- listdlg – создание окна для выбора вариантов значений параметра из списка;
- menu – создание меню диалогового ввода (choice=menu(header, item1, item2, ...);
- msgbox – создание окна сообщений;
- questdlg - создание окна запроса;
- warndlg - создание окна предупреждения;
- uigetfile - создание окна открытия файла;
- uiputfile - создание окна сохранения файла;
- uisetcolor - создание окна выбора цвета;
- uisetfont - создание окна выбора шрифта;
- pagedlg - создание окна парамеров страницы;
- printdlg - создание окна для вывода на печать;
- waitbar - создание “панели ожидания”;
- makemenu - создание структуры меню;
- menubar – установка типовых свойств для объекта MenuBar;
- btngroup - создание инструментальной панели с группой кнопок;
- btnstate – запрос статуса кнопки;
- btnpress – управление кнопкой;
- btndoun – нажатие кнопки;
- btnup – “поднятие” кнопки.

Полный список функций GUI можно изучить с помощью команды help uitools.

GUI. ПРИМЕРЫ

Пример. checkbox
```
cbh = uicontrol(fh,'Style','checkbox',...
                'String','Display file extension',...
                'Value',1,'Position',[30 20 130 20]);
```

Пример. Edit Text
```
eth = uicontrol(fh,'Style','edit',...
                'String','Enter your name here.',...
                'Position',[30 50 130 20]);
```

Пример 2. Edit Text
```
eth1 = uicontrol(fh,'Style','edit',...
                'String','Enter your name and address here.',...
                'Max',2,'Min',0,...
                'Position',[30 20 130 80]);
```

Пример 3. listbox
```
lbh = uicontrol(fh,'Style','listbox',...
                'String',{'one','two','three','four'},...
                'Value',1,'Position',[30 20 130 80]);
```

Пример 4. listbox
```
lbh = uicontrol(fh,'Style','listbox',...
                'String',{'one','two','three','four'},...
                'Max',2,'Min',0,'Value',[1 3],...
                'Position',[30 20 130 80]);
```

Пример 5. popupmenu
```
pmh = uicontrol(fh,'Style','popupmenu',...
                'String',{'one','two','three','four'},...
                'Value',1,'Position',[30 80 130 20]);
```

Пример 6. table
```
a=[1 2 3; 4 5 6; 7 8 9]
fh=figure;
th = uitable(fh,'Data',a);
```

Пример 7. button
```
pbh = uicontrol(fh,'Style','pushbutton','String','Button 1',...
                'Position',[50 20 60 40]);
```

Пример 7. button. Добавление изображения на кнопку
```
img(:,:,1) = rand(16,64);
img(:,:,2) = rand(16,64);
img(:,:,3) = rand(16,64);
pbh = uicontrol(fh,'Style','pushbutton',...
                'Position',[50 20 100 45],...
                'CData',img);
```

Пример 8. radiobutton
```
rbh = uicontrol(fh,'Style','radiobutton',...
                'String','Indent nested functions.',...
                'Value',1,'Position',[30 20 150 20]);
```

Пример 9. slider

```
sh = uicontrol(fh,'Style','slider',...
                'Max',100,'Min',0,'Value',25,...
                'SliderStep',[0.05 0.2],...
                'Position',[30 20 150 30]);
```

Пример 10. Static Text

```
sth = uicontrol(fh,'Style','text',...
                'String','Select a data set.',...
                'Position',[30 50 130 20]);
```

Пример 11. togglebutton

```
tbh = uicontrol(fh,'Style','togglebutton',...
                'String','Left/Right Tile',...
                'Value',0,'Position',[30 20 100 30]);
```

Пример 11. panel

```
ph = uipanel('Parent',fh,'Title','My Panel',...
             'Position',[.25 .1 .5 .8]);
```

Пример 12. axes

```
ah = axes('Parent',fh,'Position',[.15 .15 .7 .7]);
```

Пример 13. Установить или спрятать стандартную панель инструментов

```
set(fh,'Toolbar','figure');  % Display the standard toolbar
set(fh,'Toolbar','none');    % Hide the standard toolbar
```


Пример 14. Вызвать диалог печати

```
printdlg
printdlg(fig)
printdlg('-crossplatform',fig)
printdlg('-setup',fig)
```

Пример 15. Сохранить файл fig

```
figure;
surf(peaks);
savefig('PeaksFile.fig');
close(gcf)
```

Пример 16. Сохранить 2 картинки в файл fig

```
h(1) = figure;
z = peaks;
surf(z)

h(2) = figure;
plot(z)

savefig(h,'TwoFiguresFile.fig');
close(h)
```

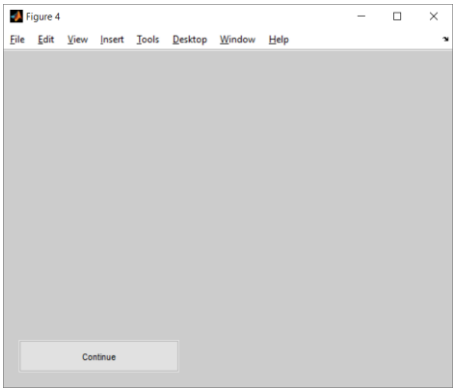
Пример. Команда saveas. Сохранить как

```
saveas(h,'filename.ext')
saveas(h,'filename','format')
```

```
h = figure;
z = peaks;
surf(z)
saveas(gcf,'logo', 'pdf')
```

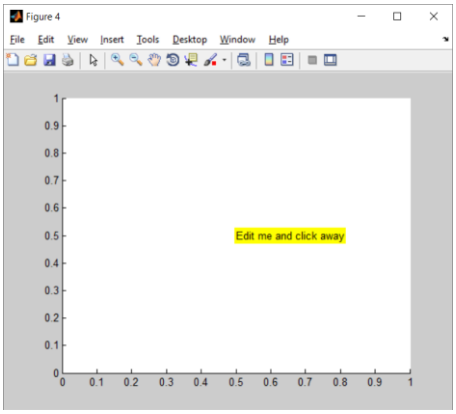
Пример. Команды uiwait и iresume.
Команда uiwait - блокирует выполнение программы и дожидается возобновления
iresume (h) - возобновляет выполнение программы, приостановленное uiwait.

```
%%
f = figure;
h = uicontrol('Position',[20 20 200 40],'String','Continue',...
              'Callback','uiresume(gcbf)');
disp('This will print immediately');
uiwait(gcf);
disp('This will print after you click Continue');
close(f);
```



Пример. waitfor -
Блокирует выполнение и ждет события или условия

```
figure;
textH = text(.5, .5, 'Edit me and click away');
set(textH,'Editing','on', 'BackgroundColor',[1 1 1]);
disp('This prints immediately.');
drawnow
waitfor(textH,'Editing','off');
set(textH,'BackgroundColor',[1 1 0]);
disp('This prints after text editing is complete.');
```



waitforbuttonpress
K = waitforbuttonpress блокирует поток выполнения вызывающего, пока функция не обнаруживает, что пользователь нажал кнопку мыши или нажал клавишу, когда окно фигуры активно. Функция возвращает

0, если он обнаруживает щелчок кнопки мыши
1, если он обнаруживает нажатие клавиши

```
fh=figure;
w = waitforbuttonpress;
if w == 0
    disp('Button click')
else
    disp('Key press')
end
```

Пример. GUI with slider (сохранить как функцию!)

```
function mygui()
    figure
    uicontrol('Style','slider','Callback',@display_slider_value);
end
function display_slider_value(hObject,eventdata)
        disp(['Slider moved to ' num2str(get(hObject,'Value'))]);
end
```

Пример. GUI с анонимной функцией (сохранить как функцию!)
Используйте массив ячеек, чтобы указать функцию обратного вызова, которая принимает входные аргументы, которые вы хотите использовать в функции. Первый элемент в массиве ячеек содержит дескриптор функции или имя функции. Другими элементами массива ячеек являются входные аргументы функции, разделенные запятыми.

```
function somegui()
    myvar = 5;
    figure

uicontrol('Style','pushbutton','Callback',{@pushbutton_callback,myv
ar});
end
function pushbutton_callback(hObject,eventdata,x)
    display(x);
end
```

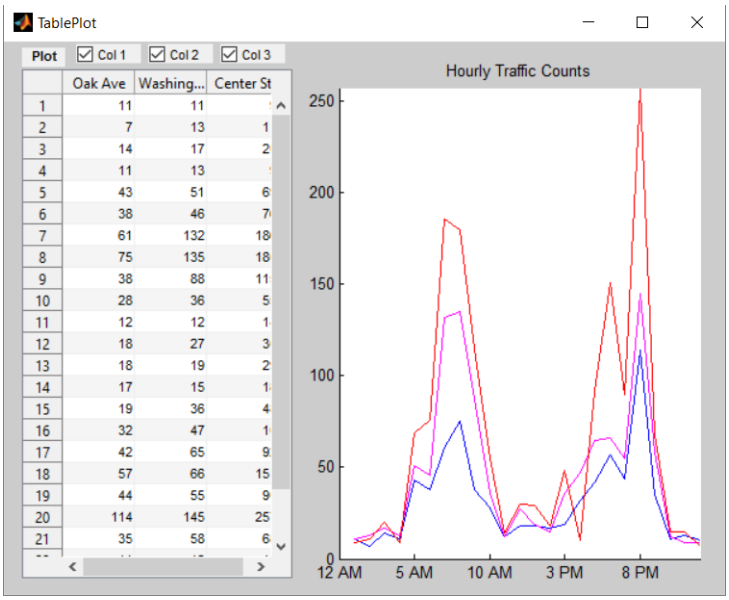

Пример. Три графика по таблице из файла

```
function varargout = tableplot(varargin)
% tableplot.m -- Programmatic main function to set up a GUI
%                containing a uitable and an axes which
%                displays columns of the table as lines,
%                plus markers that show table selections.
%
% The following callbacks are also provided, as subfunctions:
%    plot1_callback   - Plot column selected on menu as line
%    select_callback  - Plot selected table data as markers
%  Being subfunctions, they do not need handles passed to them.
%
```

```
%   Copyright 2008 The MathWorks, Inc.

% Create a figure that will have a uitable, axes and checkboxes
figure('Position', [100, 300, 600, 460],...
       'Name', 'TablePlot',...  % Title figure
       'NumberTitle', 'off',... % Do not show figure number
       'MenuBar', 'none');      % Hide standard menu bar menus

% Load some tabular data (traffic counts from somewhere)
count = load('count.dat');
tablesize = size(count);    % This example data is 24-by-3

% Define parameters for a uitable (col headers are fictional)
colnames = {'Oak Ave', 'Washington St', 'Center St'};
% All column contain numeric data (integers, actually)
colfmt = {'numeric', 'numeric', 'numeric'};
% Disallow editing values (but this can be changed)
coledit = [false false false];
% Set columns all the same width (must be in pixels)
colwdt = {60 60 60};
% Create a uitable on the left side of the figure
htable = uitable('Units', 'normalized',...
                 'Position', [0.025 0.03 0.375 0.92],...
                 'Data',  count,...
                 'ColumnName', colnames,...
                 'ColumnFormat', colfmt,...
                 'ColumnWidth', colwdt,...
                 'ColumnEditable', coledit,...
                 'ToolTipString',...
                 'Select cells to highlight them on the plot',...
                 'CellSelectionCallback', {@select_callback});

% Create an axes on the right side; set x and y limits to the
% table value extremes, and format labels for the demo data.
haxes = axes('Units', 'normalized',...
             'Position', [.465 .065 .50 .85],...
             'XLim', [0 tablesize(1)],...
             'YLim', [0 max(max(count))],...
             'XLimMode', 'manual',...
             'YLimMode', 'manual',...
             'XTickLabel',...
             {'12 AM','5 AM','10 AM','3 PM','8 PM'});
title(haxes, 'Hourly Traffic Counts')   % Describe data set
% Prevent axes from clearing when new lines or markers are plotted
hold(haxes, 'all')

% Create an invisible marker plot of the data and save handles
% to the lineseries objects; use this to simulate data brushing.
hmkrs = plot(count, 'LineStyle', 'none',...
                    'Marker', 'o',...
                    'MarkerFaceColor', 'y',...
                    'HandleVisibility', 'off',...
                    'Visible', 'off');

% Create an advisory message (prompt) in the plot area;
% it will vanish once anything is plotted in the axes.
axpos = get(haxes, 'Position');
ptpos = axpos(1) + .1*axpos(3);
ptpos(2) = axpos(2) + axpos(4)/2;
ptpos(3) = .4; ptpos(4) = .035;
hprompt = uicontrol('Style', 'text',...
                    'Units', 'normalized',...
                    'Position', ptpos,... % [.45 .95 .3 .035],...
                    'String',...
                      'Use Plot check boxes to graph columns',...
```

```
                    'FontWeight', 'bold',...
                    'ForegroundColor', [1 .8 .8],...
                    'BackgroundColor', 'w');

% Create three check boxes to toggle plots for columns
uicontrol('Style', 'checkbox',...
          'Units', 'normalized',...
          'Position', [.10 .96 .09 .035],...
          'TooltipString', 'Check to plot column 1',...
          'String', 'Col 1',...
          'Value', 0,...
          'Callback', {@plot_callback,1});
uicontrol('Style', 'checkbox',...
          'Units', 'normalized',...
          'Position', [.20 .96 .09 .035],...
          'TooltipString', 'Check to plot column 2',...
          'String', 'Col 2',...
          'Value', 0,...
          'Callback', {@plot_callback,2});
uicontrol('Style', 'checkbox',...
          'Units', 'normalized',...
          'Position', [.30 .96 .09 .035],...
          'TooltipString', 'Check to plot column 3',...
          'String', 'Col 3',...
          'Value', 0,...
          'Callback', {@plot_callback,3});

% Create a text label to say what the checkboxes do
uicontrol('Style', 'text',...
          'Units', 'normalized',...
          'Position', [.025 .955 .06 .035],...
          'String', 'Plot',...
          'FontWeight', 'bold');

% Subfuntions implementing the two callbacks
% ------------------------------------------

    function plot_callback(hObject, eventdata, column)
    % hObject     Handle to Plot menu
    % eventdata   Not used
    % column      Number of column to plot or clear

    colors = {'b','m','r'}; % Use consistent color for lines
    colnames = get(htable, 'ColumnName');
    colname = colnames{column};
    if get(hObject, 'Value')
        % Turn off the advisory text; it never comes back
        set(hprompt, 'Visible', 'off')
        % Obtain the data for that column
        ydata = get(htable, 'Data');
        set(haxes, 'NextPlot', 'Add')
        % Draw the line plot for column
        plot(haxes, ydata(:,column),...
            'DisplayName', colname,...
            'Color', colors{column});
    else % Adding a line to the plot
        % Find the lineseries object and delete it
        delete(findobj(haxes, 'DisplayName', colname))
    end
    end


    function select_callback(hObject, eventdata)
    % hObject    Handle to uitable1 (see GCBO)
    % eventdata  Currently selected table indices
```

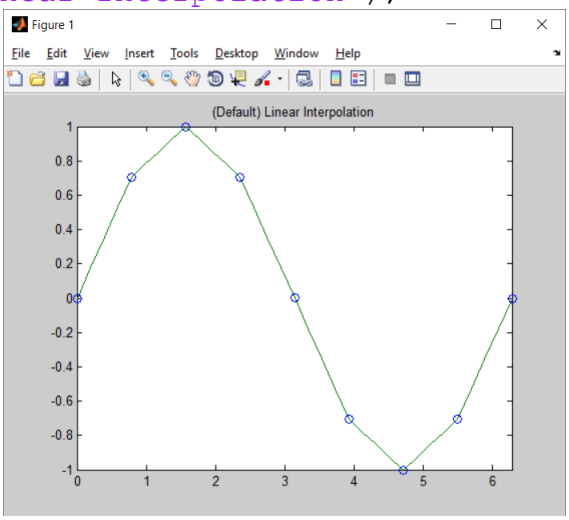
```
    % Callback to erase and replot markers, showing only those
    % corresponding to user-selected cells in table.
    % Repeatedly called while user drags across cells of the uitable

        % hmkrs are handles to lines having markers only
        set(hmkrs, 'Visible', 'off') % turn them off to begin

        % Get the list of currently selected table cells
        sel = eventdata.Indices;     % Get selection indices (row, col)
                                     % Noncontiguous selections are ok
        selcols = unique(sel(:,2));  % Get all selected data col IDs
        table = get(hObject,'Data'); % Get copy of uitable data

        % Get vectors of x,y values for each column in the selection;
        for idx = 1:numel(selcols)
            col = selcols(idx);
            xvals = sel(:,1);
            xvals(sel(:,2) ~= col) = [];
            yvals = table(xvals, col)';
            % Create Z-vals = 1 in order to plot markers above lines
            zvals = col*ones(size(xvals));
            % Plot markers for xvals and yvals using a line object
            set(hmkrs(col), 'Visible', 'on',...
                            'XData', xvals,...
                            'YData', yvals,...
                            'ZData', zvals)
        end
    end
end
```



ИНТЕРПОЛЯЦИЯ И АППРОКСИМАЦИЯ ДАННЫХ

Пример. Линейная интерполяция

```
%%
x = 0:pi/4:2*pi;
v = sin(x);
xq = 0:pi/16:2*pi;
figure
vq1 = interp1(x,v,xq);
plot(x,v,'o',xq,vq1);
xlim([0 2*pi]);
title('(Default) Linear Interpolation');
```



Пример. Сплайн интерполяция

```
figure
vq2 = interp1(x,v,xq,'spline');
plot(x,v,'o',xq,vq2);
xlim([0 2*pi]);
title('Spline Interpolation');
```