```
function varargout = plotyn(varargin)

%PLOTYN     Extends plot to create a separate Y-axius for each dataset. It

           %is a generalization of Matlab's inbuilt function PLOTYY.

%

 %Creation mode, needs to be run first:

 %Syntax:   [hax, hlines, data] = plot4y(data(

 %Syntax:   [hax, hlines, data] = plot4y(data, pos(

 %Syntax:   [hax, hlines, data] = plot4y(data, pos, parent(

%

 %Update mode, to be performed on existing axes to update their plots,

 %colors or Y-axis labels:

 %Syntax:   plot4y(hax, hlines, data(

%

 %Inputs:   data is a struct with at least two fields (X, Y(

           %The dimensionality of data determines the number of graphs

           %data(1).X      X coordinates of first dataset (Same for all(

           %data(i).Y      Y coordinates of i-th dataset

           %data(i).Color   Color of i-th dataset (Optional(

           %data(i).YLabel  Y-axis label of i-th dataset (Optional(

%

           %pos            Array of 5 numbers positioning the axes on the

                          %current figure (handle) in 'Pixel' units

%

           %parent         Handle onto which the axes are drawn

%

 %Outputs:  hax            Double array containing the axes' handles

           %hlines         Double array containing the lines' handles

           %data           Struct with the input data
```

```
%
 %Example:
            %x = 0 : 10; data(1).X = x;
            %data(1).Y = x .^ 1; data(1).YLabel = 'First Y-axis;'
            %data(2).Y = x .^ 2; data(2).YLabel = 'Second Y-axis;'
            %data(3).Y = x .^ 3; data(3).YLabel = 'Third Y-axis;'
            %data(4).Y = x .^ 4; data(4).YLabel = 'Fourth Y-axis;'
            %figure;
]            %hax, hlines, data] = plotyn(data;(
            %legend(hlines, 'y = x', 'y = x^2', 'y = x^3', 'y = x^4', 2... ,
'                  %Location', 'NorthWest;('
%
   %Now you can change the data in the graph for datasets 2 and 3
%
            %data(2).Y = x .^ (1 / 2;(
            %data(3).Y = x .^ (1 / 3;(
            %plotyn(hax, hlines, data(
%
 %See also Plot, Plotyy
%
 %Based on :    plotyyy.m by Denis Gilbert, Ph.D.
    :           %ploty4.m by Peter (PB) Bodin
 %
 %Created by modification of the aforementioned functions by:
 %Jakub Nedbal
 %April 2015
 %Distributed under BSD license.
 %Create new axes and lines within them
if isstruct(varargin{1({
 %    Get the data struct
   data = varargin{1;{
 %    If available, get the position
   if nargin > 1
       pos = varargin{2;{
 %    If position not supplied, put the axes in the middle of a new figure
   else
 %        Get position of new figure
       pos = get(gcf, 'Position;('
 %        Convert it into the position of the axes
       pos = [0.15, 0.2, 0.7, 0.6, 0.1] .* pos([3, 4, 3, 4, 3;([
   end
 %    If the position if just four points long, add the fifth one as 10%
 %    of the width of the figure
   if numel(pos) < 5
 %        For only two axes, there is no need for offsetting them
       if numel(data) < 3
           pos(5) = 0;
       else
           pos(5) = 0.1 * pos(3;(
       end
   end
 %    If the parent handle is specified, them use it
   if nargin > 2
       parent = varargin{3;{
```

```
    else
 %         Otherwise create a new figure or use the handle of the current
 %         figure
        parent = gcf;
    end
 %If existing axes are to be updated, do so
elseif ishandle(varargin{1({
    updateGraph(varargin;({:}
    return
else
    error('Check you sytnax('!
end


 %axes handle
hax = zeros(1, numel(data;((


 %lines handle
hlines = hax;


 %Number of graphs
N = numel(data;(


 %Colors
if ~isfield(data, 'Color('
 %     If no color specified, sample HSV colormap
    col = hsv(N;(
    for i = 1 : numel(data(
        data(i).Color = 0.75 * col(i;(: ,
    end
end

 %Offset of the central graph
Loff = ceil((N - 2) / 2);          % Left offset
Roff = max(0, ceil((N - 3) / 2));   % Right offset

for i = 1 : N
 %     Left offset
    Loffset = (Loff - (mod(i, 2) .* (i - 1) / 2)) * pos(5;(
 %     Right offset
    Roffset = (Roff - (mod(i + 1, 2) .* (i - 2) / 2)) * pos(5;(

 %     Make the axes invisible by setting their color to that of the parent
    if i == 3
        cfig = get(parent, 'Color;('
    end

 %     Plot one graph at a time
 %     Get its location on the parent
    Cpos = pos(1 : 4) + [Loffset, 0, -(Loffset + Roffset), 0;[

 %     Calculate the limits
    ax = axes('Parent', parent, 'Units', 'pixels', 'Position', Cpos... ,
'                 Color', 'none', 'YColor', data(i).Color;(

 %     Plot data
    hlines(i) = line(data(1).X, data(i).Y, 'Parent', ax... ,
'                     Color', data(i).Color;(

 %     Place the Y-axis on the right
    if i / 2 == round(i / 2(
        ax.YAxisLocation = 'right;'
    end
 %     Dont show any X-tick on furhter axes
    if i > 1
        ax.XTick;[] =
    end
 %     Only apply for stretched axes
    if i > 2
        ax.XColor = cfig;
 %         Determine the proper x-limits for the third and fourth axes
        scale = Cpos(3) / W;
 %         Set the X-limits accordingly
        if i / 2 == round(i / 2;(
 %             Even datasets with Y-axis on the right
            ax.XLim = [limx(1), limx(1) + scale * (limx(2) - limx(1;[((
        else
 %             Odd datasets with Y-axis on the left
            ax.XLim = [limx(2) - scale * (limx(2) - limx(1)), limx(2;[(
        end
    end
    if i == 1
 %         Switch box on
        ax.Box = 'on;'
```

```
 %        store width and Xlimits of the first axis
        W = Cpos(3;(
 %        Store the first axis limits
        limx = ax.XLim;
    end
 %    Set the Y label if defined
    if isfield(data, 'YLabel('
        ax.YLabel.String = data(i).YLabel;
    end
 %    Store the axes handle
    hax(i) = ax;
end

 %Put main axes on top;
for i = 1 : min(2, N(
    uistack(hax(i), 'top;('
end
 %return axes handles, line handles, data
varargout = {hax, hlines, data;{
end

function updateGraph(varargin(
 %Function that simply updates existing graph
%
 %Update mode, to be performed on existing axes to update their plots,
 %colors or Y-axis labels:
 %Syntax:   data = plot4y(hax, hlines, data(
 %Data           : varargin{3;{
 %Axes handles    : varargin{1;{
 %Lines handles   : varargin{2;{



 %Number of graphs
N = numel(varargin{3;({

if isfield(varargin{3}, 'Color('
    for i = 1 : N
        set(varargin{1}(i), 'YColor', varargin{3}(i).Color;(
    end
end

if isfield(varargin{3}, 'YLabel('
    for i = 1 : N
        set(get(varargin{1}(i), 'YLabel... ,('
'            String', varargin{3}(i).YLabel;(
    end
end

for i = 1 : N
    set(varargin{2}(i), 'XData', varargin{3}(1).X... ,
'                        YData', varargin{3}(i).Y;(
    if i == 1
 %        store width and Xlimits of the first axis
        W = get(varargin{1}(1), 'Position;('
        W = W(3;(
 %        Store the first axis limits
        limx = get(varargin{1}(i), 'XLim;('
    else
 %        Get current axes width
        CW = get(varargin{1}(i), 'Position;('
 %        Get scaling factor
        scale = CW(3) / W;
 %        Set the X-limits accordingly
        if i / 2 == round(i / 2;(
 %            Even datasets with Y-axis on the right
            set(varargin{1}(i), 'XLim... ,'
]                limx(1), limx(1) + scale * (limx(2) - limx(1;([((
        else
 %            Odd datasets with Y-axis on the left
            set(varargin{1}(i), 'YLim... ,'
                      ]limx(2) - scale * (limx(2) - limx(1)), limx(2;([(
        end
    end
end


end
```