```
function varargout= customGrid(varargin)

% CUSTOMGRID generates a user define grid on 2D plots

% H=customGrid([ax],'PropertyName',value) produces an arrary of custom grid

% lines defined by the property values. ax is the axes handel, if omitted,

% the current axes is used. Valid properties are 'XGrid' followed by an

% array of x coordinates of the position of the lines. 'YGrid' followed by an

% array of y coordinates of the position of the lines.

%

% For example customGrid('YGrid',[0 1]) will produce two horizontal grid

% lines at positions 0 and 1.

% Other options are 'XCurve' and 'YCurve', both of which are followed by

% cell arrays of X data and Y data. The data must be same length. This

% allows completely arbitary (ie curved) lines.

% The line format for each grid can be specfied with the following options

% 'GridAlpha' alpha transparance value (default) 0.1,

% 'Color' the line color, either an RGB triplet or char colour code, eg

% [0.15, 0.15,0.15] (the default) or 'g' for a green line.

% 'LineStyle', the line style (default is '-'), 'LineWidth' default is 0.1

% 'StoreUserData', default true, stores the handles to each line in axes

% userdata field. You can disable this if you are already using this field

% by setting it to false.

% customGrid('clear') removes all the grid lines, customGrid('clear',h)

% where h is any array of handles, removes the lines with handles h.

% This code uses patchline  by  Brett Shoelson
http://uk.mathworks.com/matlabcentral/fileexchange/36953-patchline

% Example:

% figure(1)

% clf
% t = 0:pi/64:4*pi;

% plot(t,sin(t))  %simple plot

% customGrid('YGrid',0)  %add the zero axis line

% pause()

% h=customGrid(gca,'XGrid',[0 1 5 6 7],'YGrid',[1 0.4 1.5], ...

% 'Color',[1 0 1 ],'LineStyle',':','LineWidth',2); %add some more lines

% pause()

% y{1}=cos(t);

% x{1}=sin(t);

% y{2}=0.5*cos(t);

% x{2}=0.5*sin(t);

% %now add some curved lines

% h=customGrid(gca,'XCurve',x,'YCurve',y,'Color','g','LineStyle','-','LineWidth',.1);

% pause()

% customGrid('clear',h(1:3))

% pause()

% customGrid('clear')

%

% AUTHOR Phil Birch

% DATE 2014


[REG,PROP]=parseparams(varargin);

ax=gca;

if ~isempty(REG)  %if the first param is an axes, use it

   if isa(REG{1},'matlab.graphics.axis.Axes') || isa(REG{1},'double')

       ax=REG{1};

   else
```

```
        %class(REG{1})
        error('PMB:custgrid','Unknown input type for first parameter');
    end
end

%default values
XGrid=[];
YGrid=[];
StoreUserData=true;
Color=[0.15,0.15,0.15];
GridAlpha=0.15;
LineStyle='-';
LineWidth=0.1;
XCurve=[];
YCurve=[];

%parse the parameters
for p=1:length(PROP)
    if strcmpi(PROP{p},'XGrid')
        XGrid=PROP{p+1};
    end
    if strcmpi(PROP{p},'YGrid')
        YGrid=PROP{p+1};
    end
    if strcmpi(PROP{p},'XCurve')
        XCurve=PROP{p+1};
    end
    if strcmpi(PROP{p},'YCurve')
        YCurve=PROP{p+1};
    end
    if strcmpi(PROP{p},'GridAlpha')
        GridAlpha=PROP{p+1};
    end
    if strcmpi(PROP{p},'LineStyle')
        LineStyle=PROP{p+1};
    end
    if strcmpi(PROP{p},'LineWidth')
        LineWidth=PROP{p+1};
    end
    if strcmpi(PROP{p},'Color')
        Color=PROP{p+1};
    end
    if strcmpi(PROP{p},'StoreUserData')
        StoreUserData=PROP{p+1};
    end


end
if length(XCurve)~=length(YCurve)
    error('PMB:differLenth','Xcurve and Ycurve must be same length');
end
%if clear is called test if we are using UserData, if not clear user
%supplied list of patches
hdel=[];
```

```
if ~isempty(PROP)
   if strcmpi(PROP{1},'clear')
      if length(PROP)>1 %test if next parm is a patch, if so only delete these ones
         if isa(PROP{2},'matlab.graphics.primitive.Patch')
            hdel=PROP{2};
         end
      end
      if ~StoreUserData
         delete(hdel)
         return;
      else %using storage, find the handles that match and delete them
         h=get(ax,'UserData');
         if isempty(hdel) %if this wasn't set above, delete all the lines
            delete(h);
            h=[];
            set(ax,'UserData',h);
         else  %only delete lines from the user supplied list. Search for them in userdata
            ind=[];
            for q=1:length(h)
               for qq=1:length(hdel)
                  if isequal(h(q),hdel(qq))
                     ind=[ind q];
                  end
               end
            end

            delete(h(ind));
            h(ind)=[];

            set(ax,'UserData',h);

          end

        end


        return;

    end

end

%find the extent of each line, either the axes limit or the user data

%xm=ax.XLim;

xm=get(ax,'XLim');

if ~isempty(XGrid)

    xm(1)=min(xm(1),min(XGrid));

    xm(2)=max(xm(2),max(XGrid));

end

ym=get(ax,'YLim');

%ym=ax.YLim;

if ~isempty(YGrid)

    ym(1)=min(ym(1),min(YGrid));

    ym(2)=max(ym(2),max(YGrid));

end

c=1;

for p=1:length(XGrid)


h(c)=patchline([XGrid(p),XGrid(p)],ym,'edgecolor',Color,'linewidth',LineWidth,'edgealpha',GridAlpha.
..

        ,'linestyle',LineStyle);

    c=c+1;
```

```
end

for p=1:length(YGrid)

h(c)=patchline(xm,[YGrid(p),YGrid(p)],'edgecolor',Color,'linewidth',LineWidth,'edgealpha',GridAlpha..
.
        ,'linestyle',LineStyle);

    c=c+1;

end


for p=1:length(XCurve)

h(c)=patchline(XCurve{p},YCurve{p},'edgecolor',Color,'linewidth',LineWidth,'edgealpha',GridAlpha...
        ,'linestyle',LineStyle);

    c=c+1;

end

%store the handles

horg=[];

if StoreUserData

    horg=get(ax,'UserData');

    set(ax,'UserData',[horg h]);

end

if nargout>0

    varargout{1}=[horg h];

end
```