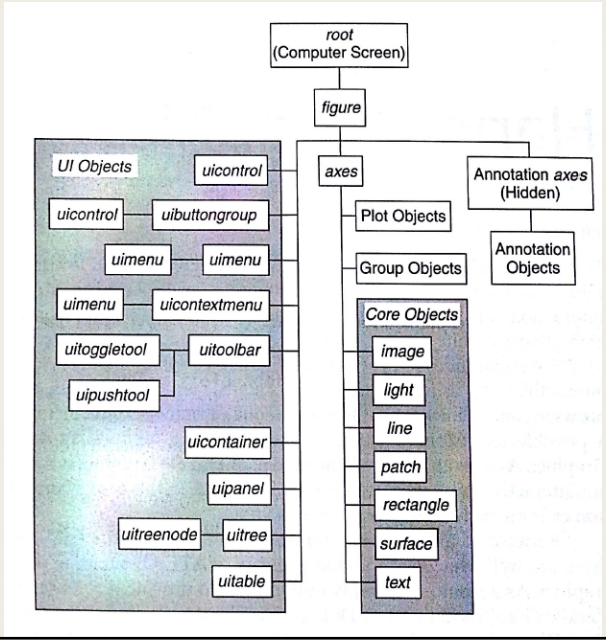
Graphics

- Vector graphics.
- Overview of common functions and parameters.
- Graphics in Matlab.

Graphics in Matlab



Graphics - Vector Images

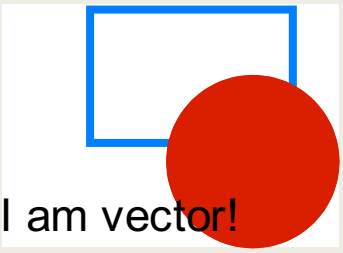
- Image composed and stored as a sequence of pre-set shapes or objects.
- Lines, rectangles, ellipses, text etc.
- Described in terms of size, position, drawing colour, fill colour.
- Each object's characteristics can be edited independently while in this graphical form.

Graphics – Vector Images

- Often called vector graphics.
- Common drawing packages allow the creation of this form of image.
- Compactly storable in files. PDF
- We will look at typical commands and file editing.

Graphics – Vector Images

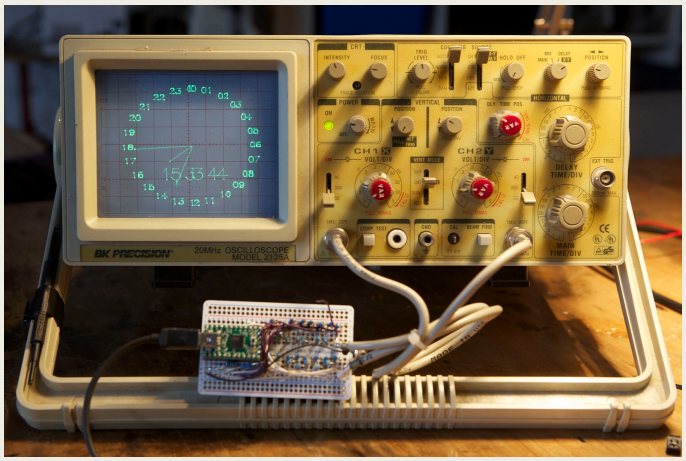
- Example of a graphic vector image created using “Autoshapes”.
- Other popular vector graphic tools are Paint shop pro, Adobe Fireworks, Photoshop.



Vector Graphics

- Is to pictures what MIDI is to sound.
- Uses lines, predefined shapes, curves and (predefined text).
- Can be very compact.
- Good for plotters.
- Converted to bitmap for monitor display.

Vector monitor?



- Used for computer graphics up through the 1970s. It is a type of CRT, similar to the oscilloscope. In a vector display, the image is composed of drawn lines rather than a grid of glowing pixels as in raster graphics.
- Vector displays do not suffer from the display artifacts of aliasing and pixelation

Matlab graphics Co-ordinate systems

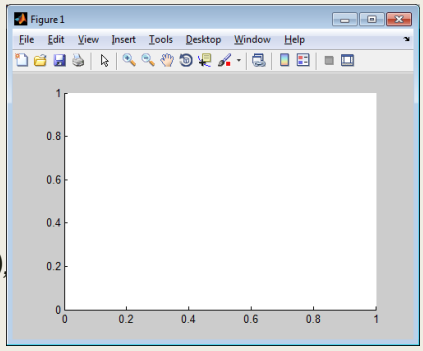
- Vector graphics based upon an x, y co-ordinate system.
- The **x** co-ordinate **runs from left to right** across the screen.
- The **y** co-ordinate usually **runs from the bottom (= 0) of the image to the top**, but sometimes from top (= 0) to the bottom.

set(gca,'XAxisLocation','top') %gca is the current axis handle

Co-ordinate systems

- To set up co-ordinate system in Matlab.
- **Haxes = axes**
  - *Sets up a co-ordinate system starting at 0 on the x- and y-axes and extending to 1.0 on the x-axis and 1.0 on the y-axis.*
- You may now draw on this system.
- You can change the axis scaling using the "axis" command.
- But by default the scaling will increase to accommodate your objects.

try: Haxes=axes('Plotboxaspectratio',[1 1 1]);

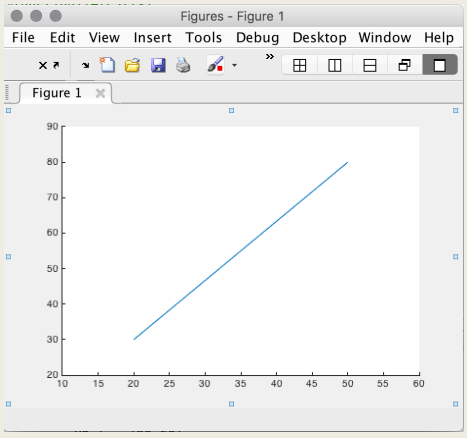


Lines in Matlab

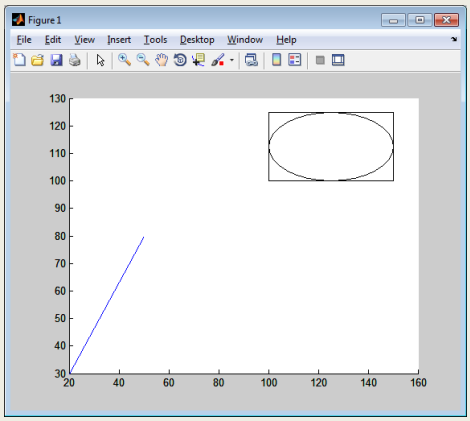
- Function line used to draw lines
- h = line(x, y) where x and y are x and y co-ordinates of the start and end of a line. h is a "handle" to the graphics object (used for setting properties).
- Example
  - *x=[20 50]*
  - *y=[30 80]*
  - *hline = line(x,y)*

Draws a line from point x=20, y=30 to the point x=50, y=80

Also try: hline=line(x,y,z);

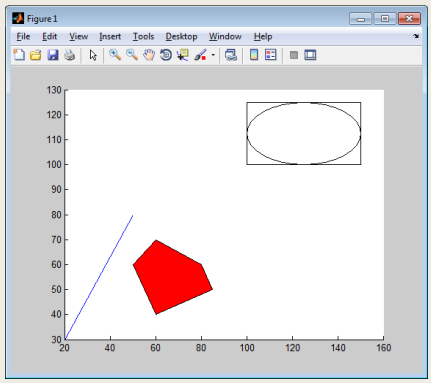


Rectangles in Matlab



- Function “rectangle” used to draw rectangles in Matlab.
- Often rectangles are defined by 2 points only
  - *Bottom left and top right.*
- In Matlab
  - *hrect = rectangle('Position', [100 100 50 25])*
    - (start position (x,y) then width and height.)
- Also used to draw ellipses and circles
  - *hcirc = rectangle('Position', [100 100 50 25], 'Curvature', [1 1]).*
    - A circle is an ellipse (with the same height as its width)

Polygons in Matlab



- h = patch(x, y, ‘r’) draws a polygon, the vertices of which are contained in x and y, and is filled by colour ‘r’.
- Example draw a red filled pentagon.
  - *Need 5 points in x and y.*
  - *x=[50 60 80 85 60]*
  - *y=[60 70 60 50 40]*
  - *hpoly = patch(x, y, ‘r’)*

Long Name	Short Name	RGB Triplet
'yellow'	'y'	[1 1 0]
'magenta'	'm'	[1 0 1]
'cyan'	'c'	[0 1 1]
'red'	'r'	[1 0 0]
'green'	'g'	[0 1 0]
'blue'	'b'	[0 0 1]
'white'	'w'	[1 1 1]
'black'	'k'	[0 0 0]

Our example so far

Drawing.m*

```
1 -    clear all
2 -    clc
3
4 -    haxes=axes;
5
6 -    x=[20 50];
7 -    y=[30 80];
8 -    hline=line(x,y);
9
10 -   hrect = rectangle('Position', [100 100 50 25]) ;
11
12 -   hcirc = rectangle('Position', [100 100 50 25], 'Curvature', [1 1]);
13
14 -   x=[50 60 80 85 60];
15 -   y=[60 70 60 50 40];
16 -   hpoly = patch(x, y, 'r');
17
18 -   Hchild=get(haxes, 'Children');
19
```

Exercises

- Draw a line from point x=20, y=30 to the point x=50, y=60.
- Draw a rectangle with the bottom left hand corner at point x=30, y=50 and the top right hand corner at point x=70, y=80.
- Draw a circle of radius 3 and centred at point x=50, y=60.
- Draw a blue filled triangle.

Handles

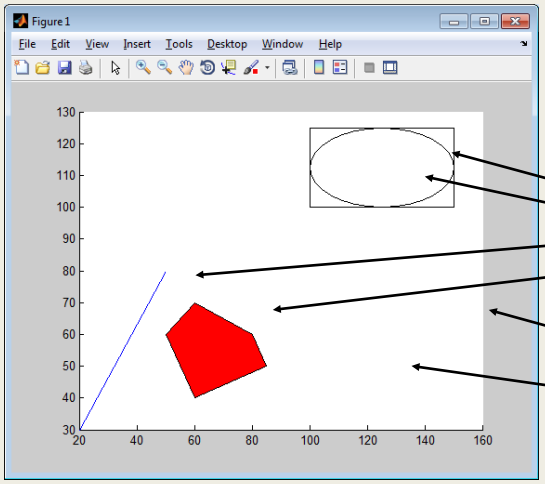
- The main figure has a “handle” in the Matlab environment.
- Handles allow Matlab to keep track of figures and graphic objects.
- Within the main figure we have an axis object; this also has a handle.
- It is a “child” object of the figure.

Handles

- We differentiate between or identify objects by their handles
- Sort of pointer.
- When we add drawing objects such as our lines, rectangles, patches, they become child objects of the axis object and are also identified by handles.
- So we now have an “axes” with four children.
- We can return there values using
- *Hchild=get(haxes, ‘Children’)*

Handles

- So our structure so far could be drawn as.



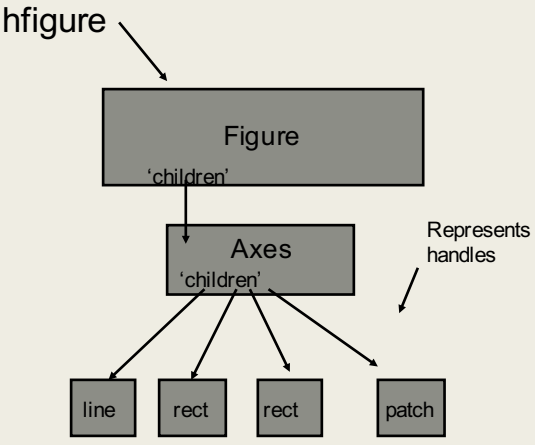
haxes has
4 children with handles
hrect,
hcirc,
hline
hpoly

Hfigure is the handle to the figure and has child object with handle

haxes

Handles

- Or in hierarchical form



Handles

- The handle of the figure is returned in variable “hfigure”. *(hfigure=figure(1))*
- It gives us access to all the properties of the figure.
  - *get(h) returns a copy of the figure’s (object’s) properties including its children.*

Handles, Try it

- Type “maindetails=get(hfigure)”
- main details lists all the properties in the figure.
- The structure includes handles to the child objects.
- We can use the handles to gain access to the child objects and alter their properties.

Properties, get() and set()

- We can retrieve a copy of the values associated with a graphic object through its handle by using S=get(hrect)
- The structure contains all the properties of the graphic object.
- However, since it is a copy we cannot change the actual information associated with the graphic object.

Properties, get() and set()

- So in true “OO”” style we must use an access method/function to adjust parameters..
- Set(h,'PropertyName',PropertyValue)
- Get(h) or Get(h, ‘PropertyName’) returns the property.
- Note ‘quotes’

```
>> s=get(hline)

          Annotation: [1x1 hg.Annotation]
        BeingDeleted: 'off'
          BusyAction: 'queue'
       ButtonDownFcn: ''
            Children: [0x1 double]
            Clipping: 'on'
               Color: [0 0 1]
           CreateFcn: ''
           DeleteFcn: ''
         DisplayName: ''
    HandleVisibility: 'on'
             HitTest: 'on'
       Interruptible: 'on'
           LineStyle: '-'
           LineWidth: 0.5000
              Marker: 'none'
     MarkerEdgeColor: 'auto'
     MarkerFaceColor: 'none'
          MarkerSize: 6
              Parent: 0.0094
            Selected: 'off'
  SelectionHighlight: 'on'
                 Tag: ''
                Type: 'line'
      UIContextMenu: []
            UserData: []
             Visible: 'on'
               XData: [20 50]
               YData: [30 80]
               ZData: [1x0 double]
```

```
>> s=get(hcirc)

          Annotation: [1x1 hg.Annotation]
        BeingDeleted: 'off'
          BusyAction: 'queue'
       ButtonDownFcn: ''
            Children: [0x1 double]
            Clipping: 'on'
           CreateFcn: ''
           Curvature: [1 1]
           DeleteFcn: ''
         DisplayName: ''
           EdgeColor: [0 0 0]
           FaceColor: 'none'
    HandleVisibility: 'on'
             HitTest: 'on'
       Interruptible: 'on'
           LineStyle: '-'
           LineWidth: 0.5000
              Parent: 0.0094
            Position: [100 100 50 25]
            Selected: 'off'
  SelectionHighlight: 'on'
                 Tag: ''
                Type: 'rectangle'
      UIContextMenu: []
            UserData: []
             Visible: 'on'
```

```
>> s=get(hpoly)

    AlphaDataMapping: 'scaled'
    AmbientStrength: 0.3000
          Annotation: [1x1 hg.Annotation]
    BackFaceLighting: 'reverselit'
        BeingDeleted: 'off'
          BusyAction: 'queue'
       ButtonDownFcn: ''
               CData: []
       CDataMapping: 'scaled'
            Children: [0x1 double]
            Clipping: 'on'
           CreateFcn: ''
           DeleteFcn: ''
    DiffuseStrength: 0.6000
         DisplayName: ''
           EdgeAlpha: 1
           EdgeColor: [0 0 0]
       EdgeLighting: 'none'
           FaceAlpha: 1
           FaceColor: [1 0 0]
       FaceLighting: 'flat'
               Faces: [1 2 3 4 5]
  FaceVertexAlphaData: []
    FaceVertexCData: []
    HandleVisibility: 'on'
             HitTest: 'on'
       Interruptible: 'on'
           LineStyle: '-'
           LineWidth: 0.5000
              Marker: 'none'
```

Order of objects

- As an drawing object is added to the axes object an entry (drawing object's handle) is placed in the "Children" array of the axis object.
- We can rearrange this array to change which object is on top.
- Again we are simply swapping handles
- We need to call "refresh" to see it.

Order of objects

- So we get the axes objects “children” array.
- hchild=get(haxes, ‘children’)
- Make a copy
  - *htemp=hchild*
- Rearrange the handles to the objects
  - *htemp(4)=hchild(1)*
  - *htemp(1)=hchild(4)*

And set the axes “children: array to our new values
set(haxes,’children’, htemp)

- or use uistack(): uistack(hpoly,'top');

Deleting objects.

- We can delete an object using its handle.
- delete(hpoly)
- Better put it back!
- hpoly = patch(x, y, ‘r’)

Stroke, Fill and Colour

- All vector graphic shapes have stroke and fill “properties”.
- They affect how the graphic is drawn.
- Stroke is how lines (and outlines are drawn) fill is how shapes are filled in.
- One property is “colour” (‘color’ in Matlab.
- Some other properties for stroke are:
  - *Width ‘LineWidth’*
  - *style (dotted dashed etc.) ‘LineStyle’*

Stroke, Fill and Colour

- To alter the fill and edge colour of a shape in Matlab:
  - *set(hrect, 'FaceColor', [1 0 0]) for fill colour*
  - *set(hrect, 'EdgeColor', [1 0 0.5]) for stroke colour.*
  - *Where hrect is a handle to the shape.*
- Line width and style may also be applied to the shape’s outline.

Transparency and “alpha” channels.

- Another property of vector graphics is the ability to add transparency.
- Many packages allow the adjustment of transparency from 0% to 100%.
- An “alpha channel” (in addition to the colour channels) is provided with the objects for this purpose.

Transparency and “alpha” channels.

- We can access the alpha channel of our shapes by the alpha property of the “patch” drawing object in Matlab.
- Face and edge (fill and stroke) have separate alpha channels.
- It has values between 0 and 1.
- An alpha value of 0 means completely transparent (i.e., invisible); an alpha value of 1 means completely opaque (i.e., no transparency).
- set(hpoly, ‘FaceAlpha’, 0.5)

Text

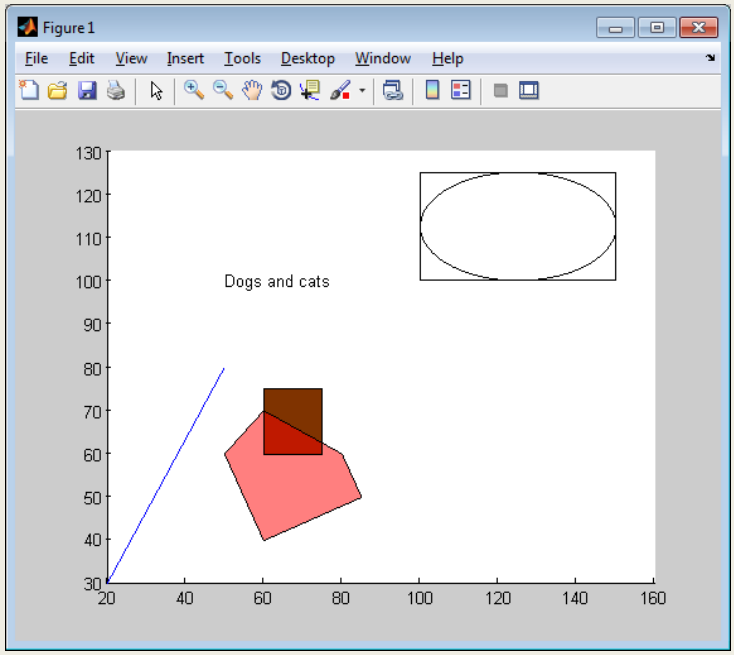
- Text may be added to vector graphics.
- htext=text(50,100, ‘Dogs and cats’)
- Properties “font” and “colour” (at least) may be changed.

Our example so far (1)

Drawing.m*

```
clc
clear all
haxes=axes;

x=[20 50];
y=[30 80];
hline=line(x,y);
hrect = rectangle('Position', [100 100 50 25]) ;
hcirc = rectangle('Position', [100 100 50 25], 'Curvature', [1 1]);

x=[50 60 80 85 60];
y=[60 70 60 50 40];
hpoly = patch(x, y, 'r');

hrect = rectangle('Position', [60 60 15 15]) ;
set(hrect, 'FaceColor', [0.5 0.2 0]) ;
set(hpoly, 'FaceAlpha', 0.5);

Hchild=get(haxes, 'Children');

htext=text(50,100, 'Dogs and cats');
```

Our example so far (2)



Exercises

- Draw the rectangle as above but change the outline to red and fill the rectangle with cyan.

Curves

- Described mathematically.
- Polynomial equations.
- Degree of equations is the highest power of x.
- Linear $y = ax + b$
  - *degree 1*
- Quadratic. $y = ax^2 + bx + c$.
  - *degree 2*
- Cubic. $y = ax^3 + bx^2 + cx + d$
  - *degree 3*

Exercises

- Linear degree 1

  e.g. $y = 2x + 5$.

  x = [-10 :10]; plot(x, 2*x + 5)

- Quadratic degree 2

  e.g. $y = x^2 + 0x + 5$.

  x = [-10 : 10]; plot(x, (x.^2) + 5)

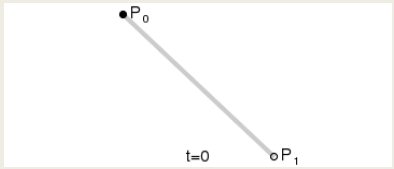
- Cubic degree 3

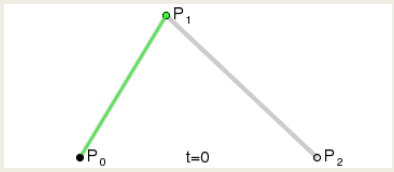
  e.g. $y = 2x^3 + 20x^2 + 3x + 2$.

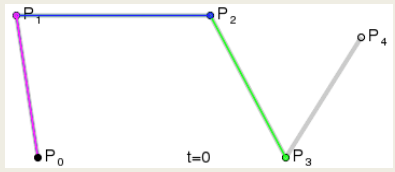
  x = [-10 : 10]; plot(x, 2*(x.^3) + 20*(x.^2) + 3*x + 2)

Bézier curve

- A Bézier curve is a parametric curve frequently used in computer graphics to model smooth curves that can be scaled indefinitely.
- Pierre Bézier first used them in the 1960's to help designing Renault cars…
- https://en.wikipedia.org/wiki/B%C3%A9zier_curve
- http://blogs.sitepointstatic.com/examples/tech/svg-curves/quadratic-curve.html



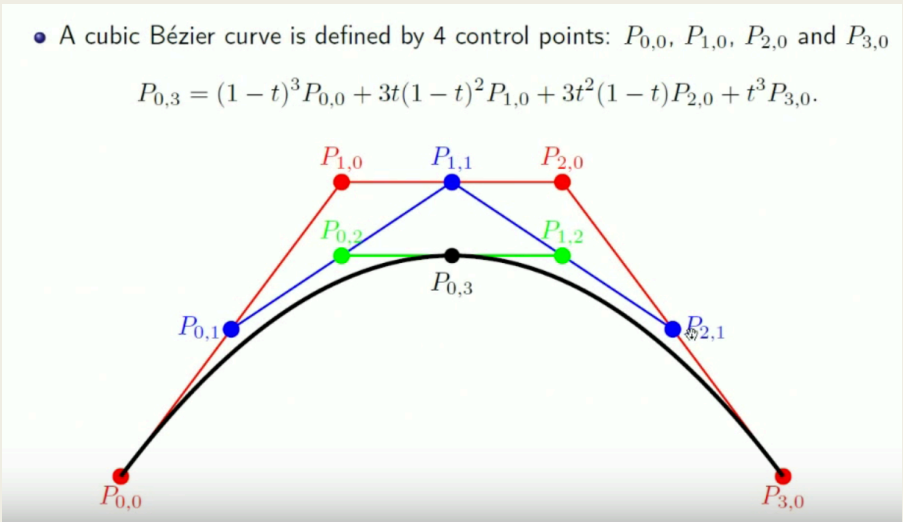




Bézier curve

- A cubic Bézier curve is defined by 4 control points: $P_{0,0}$, $P_{1,0}$, $P_{2,0}$ and $P_{3,0}$

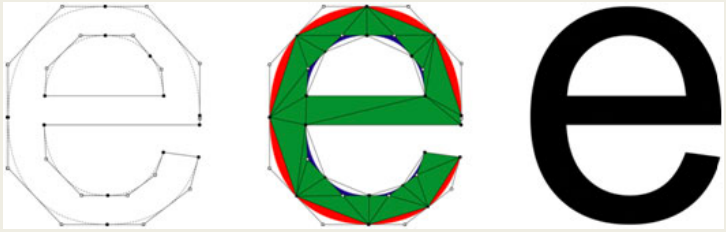
$$P_{0,3} = (1-t)^3 P_{0,0} + 3t(1-t)^2 P_{1,0} + 3t^2(1-t)P_{2,0} + t^3 P_{3,0}.$$



http://www.mathworks.com/matlabcentral/fileexchange/33828-generalised-bezier-curve-matlab-code

Bézier curve

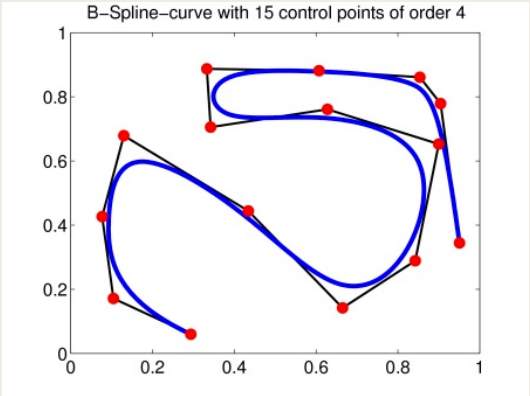
https://upload.wikimedia.org/wikipedia/commons/e/e1/HINO-4CV-01.jpg



http://guity-novin.blogspot.co.uk/2013/04/chapter-66-bezier-curves-for-digital.html

Other mathematical curves

- B-splines
- NURBS (Non-uniform rational basis spline)



http://m2matlabdb.ma.tum.de/example.jpg?MP_ID=485